technische universität
dortmund

Bachelorarbeit

Dezentrale Datenanalyse auf Netzwerken heterogener Geräte mit Blockchains

Cedric Sanders
April 2019

Gutachter:
Prof. Dr. Katharina Morik
Dr. Thomas Liebig

Technische Universität Dortmund
Fakultät für Informatik
Künstliche Intelligenz (VIII)
https://www-ai.cs.uni-dortmund.de

Inhaltsverzeichnis

Kapitel 1

Einleitung

1.1 Motivation und Hintergrund

In der Informatik gibt es sehr verschiedene Ansätze zur Verwaltung und Analyse von Daten. Klassischerweise werden die Daten dazu an einer zentralen Instanz gesammelt, um sie dann dort analysieren zu können. In einigen Fällen stellt ein solcher Ansatz jedoch nicht die geeignetste Lösung dar. Derartige zentralisierte Systeme führen meistens nicht nur zu einer geringeren Performanz, sondern sind auch anfällig für Ausfälle und Angriffe. Deshalb beschäftigt sich diese Arbeit mit einem dezentralen Ansatz zur automatisierten Verwaltung und Analyse von Daten.

Beispiele für Anwendungsgebiete, die von einer dezentralen Verwaltung und Analyse von Daten profitieren könnten sind Sensornetzwerke und die Verwaltung von Verkehrsdaten. Diese beiden Beispiele haben eine weitere Gemeinsamkeit, die eine große Rolle für das in dieser Arbeit behandelte Thema spielt. In beiden Fällen handelt es sich um Netzwerke von heterogenen Geräten. Also Geräte, die sich in ihren Eigenschaften, wie zum Beispiel Rechenleistung, Speicherkapazität, Funktion und Stromverbrauch, stark unterscheiden können und somit eine sehr unterschiedliche Handhabung erfordern.
Zusätzliche Herausforderungen entstehen auch aus der Vielseitigkeit der möglichen Anwendungsgebiete, da diese mit sehr unterschiedlichen Prioritäten verbunden sein können. Während in einigen Bereichen Sicherheit die zentrale Position einnimmt, sollte es zum Beispiel um die Analyse sensibler Patientendaten in einem Krankenhaus gehen, so können in anderen Fällen der Speicherverbrauch oder die benötigte Rechenleistung die zentralen Qualitätsmerkmale der Anwendung sein.

Möchte man die Daten in einem solchen Netzwerk nun verwalten und analysieren steht man vor einigen Problemen. Wie kann man die Datenstände der verschiedenen Geräte konsistent halten und wie kann man die benötigten Datenmengen überhaupt in einem de-

zentralisierten Netzwerk speichern?
Eine Technologie, die eine potenzielle Lösung für diese Probleme darstellt, sind Blockchains. Dabei handelt es sich um Datenblöcke, die Transaktionen, einen Zeitstempel und den Hashwert eines vorherigen Blocks enthalten, sodass sie untrennbar miteinander verbunden sind. Um eine Blockchain um weitere Blöcke zu erweitern, braucht es außerdem ein sogenanntes Consensus Verfahren. Also einen Weg wie sich die Netzwerkteilnehmer auf einen neuen Block einigen, der der Blockchain hinzugefügt wird. Die aktuell verbreiteten Consensus Verfahren stoßen in einem Netzwerk dieser Art jedoch auf einige Probleme, da sie durchgehende Aktivität und große Rechenleistungen erfordern, die leistungsschwache Geräte schlichtweg nicht aufbringen können.
Das Thema lässt sich damit also in das neue Forschungsgebiet der Ubiquitous Knowledge Discovery [12] einordnen. Dieses Gebiet vereint nicht nur die Bereiche Maschinelles Lernen und Data Mining, sondern verbindet diese auch mit den neuen Herausforderungen, die aus dem Internet of Things erwachsen, also mobile und verteilte Systeme mit sehr unterschiedlichen Eigenschaften.

1.2 Zielsetzung und Fragestellung

Ziel dieser Arbeit ist es zu untersuchen, ob Blockchains geeignet sind, um Daten in dezentralen Netzwerken aus heterogenen Geräten zu verwalten und zu analysieren. Zu diesem Zweck gilt es die verschiedenen Umsetzungen der Blockchain zu ergründen, die sich aus den unterschiedlichen Consensus Verfahren ergeben. Durch entsprechende Experimente sollen dann Statistiken ermittelt werden, die dabei helfen sollen die generelle Nutzbarkeit der Verfahren zu beurteilen beziehungsweise sich je nach gegebenem Anwendungsfall für eines der Verfahren zu entscheiden. Aus dieser Zielsetzung ergeben sich einige Fragen, die im Rahmen dieser Arbeit beantwortet werden sollen.

Wie muss ein Consensus Verfahren aussehen, das den Anforderungen an Rechenleistung, Speicherplatz und Energieverbrauch genügt, welche aus einem dezentralisierten Netzwerk heterogener Geräte entstehen? Gibt es Probleme, welche die aktuellen Verfahren ungeeignet machen? Wie genau sehen diese Probleme aus und wie kann man diese Verfahren eventuell abändern, um das gewünschte Ergebnis zu erzielen? Welche Anforderungen und Herausforderungen stellen die aktuellen Verfahren an heterogene Geräte?

1.3 Arbeitsablauf und Evaluierung

Im Rahmen der Arbeit gilt es zunächst einmal Informationen zur Funktionsweise der Blockchain und der verschiedenen Consensus Verfahren zu sammeln. Im Anschluss daran müssen diese analysiert werden, um mögliche Herausforderungen, Kritikpunkte und vorteilhafte Ei-

genschaften herauszuarbeiten. Nachdem diese Vorarbeiten abgeschlossen sind, können die Blockchain, sowie die verschiedenen Consensus Verfahen implementiert werden.
Nun gilt es Experimente durchzuführen und Daten zu sammeln, die Aufschluss darüber geben können, ob die Verfahren für die gegebene Aufgabenstellung geeignet sind. Abschließend müssen diese Ergebnisse dann evaluiert werden, um Probleme und Verbesserungsmöglichkeiten zu identifizieren und zu einer abschließenden Bewertung zu gelangen.
Die Evaluierung basiert dabei auf der Analyse einiger zentraler Qualitätsmerkmale der Netzwerkteilnehmer. Dabei handelt es sich um Kommunikationskosten, Speicherauslastung, Laufzeiten und Zuverlässigkeit. Diese Merkmale ermöglichen eine Bewertung der verschiedenen Consensus Verfahren. Außerdem können sie mit den Eigenschaften von heterogenen Geräten abgeglichen werden, um festzustellen, ob das entsprechende Verfahren auf diesem Gerät implementiert werden kann.

1.4 Aufbau der Arbeit

Im folgenden zweiten Kapitel wird zunächst die Literatur vorgestellt, welche als Grundlage dieser Arbeit dient. Die Funktionsweise der Blockchain wird erläutert, und es werden einige Arbeiten vorgestellt, die sich mit möglichen Erweiterungen und Verbesserungen der Technologie befassen. Zu diesem Zeitpunkt findet allerdings noch keine Wertung der präsentierten Konzepte und Technologien statt. Die Analyse der Eignung der Konzepte für den gewünschten Zweck findet erst im dritten Kapitel statt. Hier soll der Fokus ganz auf den verschiedenen Stärken und Schwächen sowie auf dem Vergleich der verschiedenen Verfahren liegen. Es werden Probleme sowie das Potenzial der Technologien beim Einsatz in dezentralen Netzwerken heterogener Geräte herausgearbeitet. Im vierten Kapitel wird die Implementierung der verschiedenen Verfahren thematisiert, und es werden dabei aufgetretene Probleme, sowie Designentscheidungen herausgestellt. Dazu werden zunächst einmal Anforderungen an die Funktionsweise des fertigen Programms aufgelistet. Der letzte Abschnitt befasst sich mit einem möglichen Anwendungsfall und stellt ein entsprechendes Datenanalyseverfahren vor. Das folgende fünfte Kapitel befasst sich mit den durchgeführten Experimenten und präsentiert die Ergebnisse der Arbeit mit einem entsprechenden Fazit. Es beginnt mit der Beschreibung des Versuchsaufbaus inklusive der verwendeten Datengrundlage und stellt dann die erhaltenen Ergebnisse graphisch dar. Das abschließende sechste Kapitel dient dazu einen Ausblick für das Thema zu geben und mögliche Weiterführungen der Arbeit anzuregen.

Kapitel 2

Literaturüberblick

2.1 Mathematische Grundlagen

Im Folgenden werden kurz einige der mathematischen Konzepte angerissen, die im Kontext der weiteren Arbeit eine zentrale Rolle einnehmen.

2.1.1 Hashfunktionen und Kryptographische Hashfunktionen

Hashfunktionen sind Funktionen, die sehr effizient Eingabewerte beliebiger Länge auf Ausgabewerte fester Länge abbilden. Hashfunktionen sind dabei Einwegfunktionen. Sie können also nur in eine Richtung berechnet werden. Die einzige Möglichkeit eine Hashfunktion umzukehren besteht darin die Hashfunktion auf zufällige Eingabewerte anzuwenden, bis der gewünschte Ausgabewert zurückgegeben wird.
Gilt darüber hinaus, dass unterschiedlichen Eingabewerten nicht der gleiche Ausgabewert zugewiesen wird, so ist die Hashfunktion kollisionssicher und man spricht auch von einer sogenannten Kryptographischen Hashfunktion.
Das Ergebnis einer Hashfunktion wird als Hashwert, oder auch kurz Hash, bezeichnet.

Um den Hashwert zu berechnen, wird der Eingabetext für gewöhnlich in seine einzelnen Bits zerlegt und dann mit entsprechend einfachen Operationen, wie AND, OR, XOR und LSHIFT beziehungsweise RSHIFT verändert. Welche Operationen wie oft, und in welcher Reihenfolge zum Einsatz kommen hängt ganz von der jeweiligen Hashfunktion ab.

Eingabewert	Hashwert
Testtext	`25679f2edadb95d04f92972276131c00dd7b2eae3b94a6b98f60c15a38a18af6`
lorem ipsum	`5e2bf57d3f40c4b6df69daf1936cb766f832374b4fc0259a7cbff06e2f70f269`
Input1	`b9e2b8cc6757ea3f121ef83cb88a813c329c435751b0abd19308fa19f75460b1`

2.1.2 Verifiable Random Functions

Verifiable Random Functions sind ein Konzept aus der asynchronen Kryptographie. Ziel ist es eine Funktion auszuführen, deren Ergebnis sowohl für den Ausführenden, als auch für alle anderen Parteien unvorhersehbar ist. Gleichzeitig soll der Ausführende aber in der Lage sein zu beweisen, dass die auf diesem Weg erhaltenen Werte korrekt sind.
Verifiable Random Functions lassen sich mit verschiedenen kryptographischen Verfahren konstruieren. Ein Beispiel ist der Ansatz von [5] mit der Eliptischen-Kurven-Kryptographie. Der grundsätzliche Aufbau ist allerdings immer derselbe, wie beschrieben in [14].
Man benötigt zunächst einen Generator, der ein Schlüsselpaar für die Teilnehmer generiert.

2.1 Definition (Generator). Generiert ein Schlüsselpaar.
Input: -
Output: öffentlicher Schlüssel(PK), privater Schlüssel(SK)

Mit seinem privaten Schlüssel(SK) und dem öffentlich bekannten Seed(x) ist ein Nutzer nun in der Lage die Zufallsfunktion auszuführen.

2.2 Definition (Zufallsfunktion). Ermittelt einen unvorhersehbaren Wert.
Input: privater Schlüssel(SK), Seed(x)
Output: Wert(v), Proof(p)

Durch die Veröffentlichung des erhaltenen Proofs(p) ist nun jeder in der Lage mit Hilfe des Verifikators zu überprüfen, dass der Wert(v) tatsächlich das Ergebnis der Zufallsfunktion bei der Eingabe des Seeds(x) ist.

2.3 Definition (Verifikator). Überprüft die Korrektheit eines Wertes(v)
Input: öffentlicher Schlüssel(PK), Seed(x), Wert(v), Proof(p)
Output: Ja, falls es sich um ein gültiges Ergebnis der Verifiable Random Function handelt, andernfalls Nein

2.2 Begriffserklärungen

Der folgende Abschnitt dient dazu einige Begriffe zu erklären, welche regelmäßige Anwendung im Kontext von Blockchains finden. Die Begrifflichkeiten werden teilweise im folgenden Kapitel aufgegriffen und näher erläutert.
Das Erstellen neuer Datenblöcke zur Erweiterung der Blockchain wird auch als **Mining** bezeichnet, da ein Netzwerkteilnehmer rechnerisch harte Arbeit leistet, um sich eine Belohnung zu verdienen. Diese Belohnung nennt man auch **Block Reward**, während die Komplexität der Arbeit als **Mining Complexity** bezeichnet wird. Der Wettkampf mehrerer Netzwerkteilnehmer als erstes einen neuen Block zu erstellen wird dementsprechend auch als **Mining Race** bezeichnet. Der erste Block innerhalb einer Blockchain trägt den

Namen **Genesis Block**. Ein sogenannter **Branch** entsteht, wenn zwei Netzwerkteilnehmer zur gleichen Zeit einen neuen Block erstellen und zwei alternative Versionen der Blockchain entstehen. Mit **Kryptowährungen** bezeichnet man Währungen, die nur in digitaler Form vorliegen und dementsprechend keinen materiellen Gegenwert besitzen. Der Name wird dabei von der ihnen zu Grunde liegenden Kryptographie abgeleitet.

2.3 Blockchain

Blockchains rückten erstmals im November 2008 mit der Veröffentlichung des White Papers „Bitcoin: A Peer-to-Peer Electronic Cash System“ von Satoshi Nakamoto in den Fokus [16]. Bei der Blockchain handelt es sich um eine Datenstruktur, welche aus kleineren Elementen, den Blöcken, zusammengesetzt wird. Ein Block besteht dabei aus drei zentralen Bestandteilen.

1. Nutzdaten[1]: Beinhaltet die tatsächlichen Daten (z.B. Überweisungen oder aber Sensordaten)
2. Zeitstempel: Dient dazu die Blöcke in eine zeitliche Abfolge zu bringen
3. Hashwert: Hashwert des vorhergegangenen Blocks

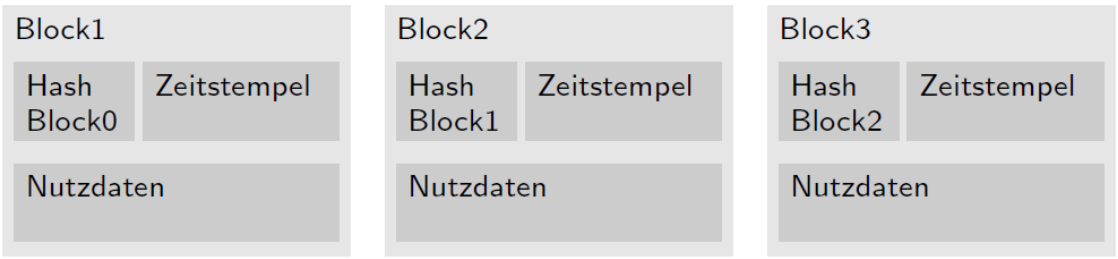

Abbildung 2.1: Beispielaufbau einer Blockchain

Jeder Block enthält also den Hashwert des vorhergegangenen Blocks, welcher wiederum die Hashwerte der ihm vorhergegangenen Blöcke beinhaltet. Sollte nun einer dieser Blöcke nachträglich editiert werden, lässt sich dies leicht an den folgenden Blöcken ablesen, da sein Hashwert nicht mehr mit dem im nachfolgenden Block eingetragenen übereinstimmt. Um die Datenstruktur in einem dezentralen Netzwerk einsetzen zu können, gilt es jedoch auch ein entsprechendes Consensus Verfahren zu implementieren.

2.4 Consensus

Als Consensus werden im Zusammenhang mit Blockchians Verfahren bezeichnet, die beschreiben wie sich Netzwerkteilnehmer auf neue Blöcke einigen, welche der Blockchain

[1] Werden auf Grund der engen Verbindung zu Kryptowährungen oft als Transaktionen bezeichnet.

hinzugefügt werden. Nakamoto beschreibt in seiner Arbeit zu Blockchains Proof of Work, welches auch heute noch bei Bitcoin im Einsatz ist, als ein solches Verfahren.
Inzwischen wurden bereits etliche weitere Verfahren wie zum Beispiel Proof of Burn [17], Proof of Luck [15], Proof of Stake [9] und Proof of Authority entwickelt. Ethereum eine dezentralisierte Plattform, die es Entwicklern ermöglicht Blockchains auf einer vorhandenen Infrastruktur zu implementieren, nutzt eine leicht abgewandelte Form des ursprünglichen Proof of Works, die zusätzliche große Anforderungen an den Speicherplatz der beteiligten Geräte stellt [2] [19].

Entwicklungen, die großes Potenzial haben, einige der bisherigen Probleme mit den gängigen Consensus Verfahren zu lösen, sind sicherlich Distributed Proof of Work von Cicada [3] und Practical Proof of Kernel Work von Xain [11].
Bei Cicada[2] handelt es sich um eine Gruppierung von Programmierern, die es sich zur Aufgabe gemacht haben nach dem Vorbild des Romans „The Jasmine Wars “ von Daniel Jeffries ein System für eine dezentrale direkte Demokratie zu entwickeln.
Der Ursprung von Xain ist eine Gruppe von Forschern aus den Bereichen maschinelles Lernen, Mathematik und Kryptographie von der University of Oxford sowie dem Imperial College in London. Xains Fokus liegt auf der Verbesserung von Blockchains mittels Reinforcment Learning und der Entwicklung ihres eigenen Consensus Verfahrens. Sie arbeiten eng mit Porsche zusammen und haben bereits erfolgreich eine Blockchain in den neusten Porsche Modellen implementiert.

Die folgenden Abschnitte dienen entsprechend dazu die vielversprechendsten dieser Verfahren näher zu erläutern. Hierbei liegt der Fokus zunächst einmal auf der reinen Funktionsweise, wie sie von den jeweiligen Autoren beschrieben wird. Die Tatsächliche Analyse der Vor- und Nachteile sowie die Eignung der Konzepte für die in dieser Arbeit angedachten Zwecke geschieht dann in Kapitel drei.

2.4.1 Proof of Work

Wie bereits beschrieben stellt Proof of Work das ursprüngliche Konzept für ein Consensus Verfahren dar, wie es von Satoshi Nakamoto zusammen mit der Blockchain vorgestellt wurde [16]. Die Idee hinter Proof of Work ist, dass sich ein Netzwerkteilnehmer das Recht verdient einen neuen Block zu Veröffentlichen. Dies tut er, indem er beweist, dass er Arbeit in Form von Rechenleistung investiert hat, um diesen neuen Block zu erstellen. Dieser Beweis lässt sich erbringen, indem die Veröffentlichung eines Blocks das Lösen eines entsprechend komplexen mathematischen Problems voraussetzt. Klassischerweise gilt es zum Beispiel einen Nonce[3] zu finden, welcher in Kombination mit dem neuen Block einen Has-

[2] Muss mit Vorsicht betrachtet werden, da die Arbeit sehr politisch und reißerisch ist.
[3] Beliebige Natürliche Zahl

hwert ergibt, der auf eine bestimmte Anzahl an Nullen endet.
Jeder Netzwerkteilnehmer sendet nun also seine Nutzdaten an alle anderen Netzwerkteilnehmer, da er nicht mit Sicherheit vorhersagen kann, wer den nächsten Block veröffentlichen wird. Alle Netzwerkteilnehmer, die einen neuen Block veröffentlichen wollen, müssen diese Daten nun sammeln und zu einem neuen Block zusammenfügen. Sobald dies geschehen ist, können sie beginnen nach einem Nonce zu suchen, der die Bedingungen erfüllt. Wer das Problem am schnellsten löst darf den neuen Block veröffentlichen.
Es kann auch vorkommen, dass mehrere Netzwerkteilnehmer zum gleichen Zeitpunkt das Problem lösen. In diesem Fall entstehen sogenannte Branches, also alternative Versionen der Blockchain. Diese werden bei Proof of Work für einige weitere Durchläufe toleriert, bis sich irgendwann ein Branch hervortut, der eine längere Blockchain aufweist als seine Konkurrenten.

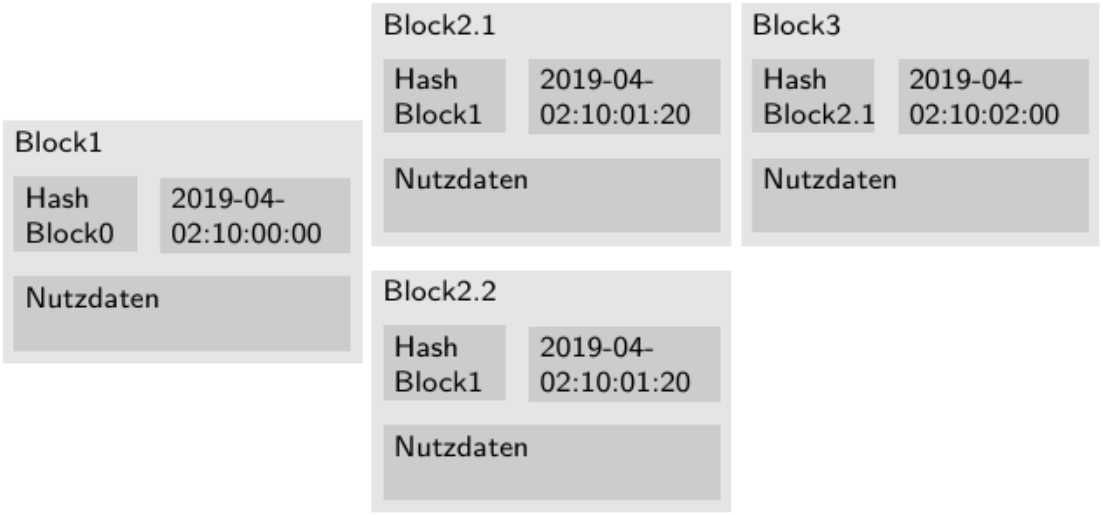

Abbildung 2.2: Auftreten eines Branches

Da dieser längste Branch das größte Investment an Arbeit und Rechenleistung repräsentiert, gilt er von da an als korrekt, und die anderen Branches werden gelöscht.
Grundsätzlich müssen sich Netzwerkteilnehmer nicht am Mining beteiligen. Im Falle von Kryptowährungen werden sie allerdings für das Veröffentlichen eines neuen Blocks belohnt. Diese Belohnung besteht einerseits aus einer festgelegten Menge an Währung, die jeder Blockersteller erhält, und andererseits aus den Bearbeitungsgebühren, die jeder Netzwerkteilnehmer bezahlt, wenn er möchte, dass Daten in die Blockchain aufgenommen werden.

Im Folgenden einen kurzes Beispiel zum Mining eines neuen Blocks. Der Nonce wird mit dem Hashwert des vorhergegangenen Blocks kombiniert und dann wird ein neuer Hashwert

berechnet. Sobald der Hashwert eine bestimmte Bedingung erfüllt, wie zum Beispiel die ersten drei Zeichen sind eine Null, gilt das Proof of Work als erfolgreich abgeschlossen, und der neue Block kann an die anderen Netzwerkteilnehmer versandt werden. Diese können dann ebenfalls die Hashfunktion ausführen und die Gültigkeit des Hashwerts verifizieren.

Nonce	Hashwert
0	`76258df8440bee5e0acaf393487be18a5e2b03e011ff50f3ebdd3efa508a1796`
1	`1bdcf2411f2828a6364df7345aa17b8ba0325d5819cb6713689e8b10aa9795b2`
2	`abc7fda919bc2941b9025f5b6b0051d0eac15909743da21e56f2a16cb1db724a`
3	`27874d4968da30133cc845441b3c64ce28c89c81276f7dd262a22d915c3d5089`
4	`b786d81904e7f99a2ade11d5014b3f03fe67bccd9ab20ae6fd4c41bd42571274`
2355	`c49c56d978ca5e93eff01f54b209305a62465d90e4c7157e3e91176231c5337b`
2356	`b8dcdcb7e29fdf927fa8f937ed99001ef045b7a170f27a10950514b2c18e7947`
2357	`000e0f58ad0044d66206c07ff90df5e72c08616cd54d128071eb75b4481def2f`

2.4.2 Proof of Stake

Proof of Stake ist ein Konzept von Sunny King und Scott Nadal, welches sie in Zusammenhang mit der Bitcoin Alternative PPCoin im August 2012 einführten [9]. Proof of Stake verfolgt einen sehr unterschiedlichen Ansatz, der deutlich stärker an die Verwendung von Blockchains für Kryptowährungen gebunden ist. Hier muss ein Netzwerkteilnehmer keinen Beweis für seine erbrachte Arbeit bringen, sondern beweisen, dass er einen entsprechenden Anteil der Währung besitzt.
Zu diesem Zweck verwendet Proof of Stake ein Konzept namens Coin Age. Coin Age beschreibt den Besitz von Währung über Zeit. Verfügt ein Netzwerkteilnehmer also über zehn Zeiteinheiten hinweg über hundert Währungseinheiten, so verfügt er über ein Coin Age von tausend. Das Coin Age wird verbraucht, sobald die zugehörigen Währungseinheiten in irgendeiner Form ausgegeben werden.
Möchte ein Netzwerkteilnehmer nun einen neuen Block erstellen, so führt er eine spezielle Transaktion aus, mit der er sich selbst Währungseinheiten überweist, und somit ihr Coin Age verbraucht, ohne die Währungseinheiten selbst zu verbrauchen. Durch diese Transaktion verdient er sich eine Erleichterung der Arbeit an einem neuen Block. Ähnlich wie bei Proof of Work gilt es jedoch auch hier noch einen vorher definierten Hashwert zu berechnen. Die Menge der möglichen Hashwerte ist hier jedoch kein fester Wert für alle Netzwerkteilnehmer, wie bei Proof of Work, sondern hängt von dem verbrauchten Coin Age ab. Somit ergibt sich eine deutlich verringerte Komplexität im Vergleich zu Proof of Work.
Auch die Auswahl der gültigen Blockchain im Falle von Branches ändert sich bei Proof of Stake. Indikator für die Gültigkeit einer Chain ist nun nicht mehr die Länge der Chain, sondern das akkumulierte Coin Age, welches innerhalb der Chain verbraucht wurde.

Um sich das ganze leichter vorstellen zu können, können wir die statische Regel „Der Hashwert muss mit drei aufeinanderfolgenden Nullen beginnen“ aus dem Beispiel von Proof of Work durch eine dynamische ersetzen. Diese könnte zum Beispiel lauten: Der Hashwert muss mit $max(0, 3 - n)$ aufeinanderfolgenden Nullen beginnen, wobei n der Menge an investiertem Coin Age entspricht.

2.4.3 Distributed Proof of Work

Das neue Consensus Verfahren von Cicada trägt den Titel Distributed Proof of Work und beruht grundsätzlich auf Proof of Work [3]. Es ist weiterhin notwendig ein mathematisches Problem zu lösen, um sich das Recht auf die Veröffentlichung des nächsten Blocks zu verdienen.
Distributed Proof of Work setzt beim Zugang zum klassischen Proof of Work an. Es können sich nicht länger zu jeder Zeit alle Netzwerkteilnehmer am Mining beteiligen. Stadtessen werden vor jedem neuen Block einige Netzwerkteilnehmer zufällig ausgewählt, welche dann für die Veröffentlichung dieses Blocks zuständig sind. Die Auswahl für einen zukünftigen Block muss nicht unmittelbar vor dessen Veröffentlichung geschehen, sondern kann auch bereits einige Blöcke früher durchgeführt werden. Außerdem werden einige zusätzliche Netzwerkteilnehmer selektiert, die im Falle von Ausfällen einspringen können. Auf diesem Weg soll sichergestellt werden, dass die Veröffentlichung des nächsten Blocks nicht zu lange dauert.
Die ausgewählten Netzwerkteilnehmer werden nun zu Gruppen zusammengestellt, welche dann gemeinsam am Proof of Work arbeiten. Die einzelnen Gruppen konkurrieren untereinander. Innerhalb der Gruppen überprüfen sich die Netzwerkteilnehmer gegenseitig, um sicherzustellen, dass niemand versucht die Gruppe zu hintergehen.
Im Falle von Distributed Proof of Work ist jeder Netzwerkteilnehmer dazu gezwungen sich am Mining zu beteiligen. Auch hier können jedoch weiterhin Belohnungen ausgeschüttet werden.

2.4.4 Practical Proof of Kernel Work

XAINs Ansatz Practical Proof of Kernel Work beruht ebenfalls auf Proof of Work und erweitert dieses durch eine Zugangsbeschränkung [11]. Die verwendeten Verfahren, um dieses Ziel zu erreichen, sind dabei allerdings vielseitiger und komplexer als bei Distributed Proof of Work.
Die Teilnahme am Proof of Work wird durch drei verschiedene Mechanismen gesteuert. Der erste dieser Mechanismen ist eine Whitelist, mit welcher man Netzwerkteilnehmer als vertrauenswürdig kennzeichnen kann. Die zweite Möglichkeit der Zugangsbeschränkung ist eine Menge von dynamischen Regeln. Diese können individuell und je nach Anwendung zusammengestellt werden. Eine solche Regel könnte es zum Beispiel dem Ersteller des letz-

ten Blocks verbieten sich an der Erstellung der nächsten drei Blöcke zu beteiligen. Bei dem letzte Mechanismus handelt es sich um einen Algorithmus zur Auswahl von zufälligen Netzwerkteilnehmern aus der Whitelist.
Die Auswahl funktioniert dabei mit Hilfe eines kontinuierlich generierten Seeds, welcher in der Blockchain eingebettet wird. Dieser Seed kann dann von den verschiedenen Netzwerkteilnehmern verwendet werden, um mit Hilfe einer Verifiable Random Function eine Lotterie durchzuführen, die über ihre Teilnahme entscheidet.
Dieses Verfahren garantiert nicht nur, dass die Auswahl von Netzwerkteilnehmern unvorhersehbar ist, sondern sorgt auch dafür, dass nur die ausgewählten Netzwerkteilnehmer wissen, dass sie ausgewählt wurden. Ein Angreifer ist also nicht in der Lage gezielt Netzwerkteilnehmer zu kompromittieren, die garantiert an der Erstellung der folgenden Blöcke beteiligt sein werden. Gleichzeitig kann ein Netzwerkteilnehmer aber zu jeder Zeit beweisen, dass er ausgewählt wurde, sollte dies jemals notwendig sein.

Kapitel 3

Analyse der Consensus Verfahren

3.1 Proof of Work

Wie bei der initialen Vorstellung der verschiedenen Consensus Verfahren, soll auch die Analyse bei Proof of Work beginnen. Als das Älteste der hier vorgestellten Verfahren gehört es auch zu den in der Literatur am meisten behandelten. Die anderen in Kapitel zwei vorgestellten Verfahren wurden alle entwickelt, um Probleme zu lösen, die innerhalb von Proof of Work aufgefallen sind.
Um Proof of Work analysieren zu können, müssen zunächst einmal die Umstände und die zugrundeliegende Motivation betrachtet werden. Proof of Work wurde entwickelt, um eine dezentrale Kryptowährung umzusetzen. Dementsprechend gehören Sicherheit, Ausfallsicherheit und Dezentralität zu den wichtigsten Zielen von Proof of Work, denen etwaige weitere Eigenschaften untergeordnet wurden. Als erstes gilt es also zu klären wie sicher Proof of Work eigentlich ist und welche Opfer eventuell gebracht werden, um diese Sicherheit zu erreichen.
Satoshi Nakamoto beschreibt in seiner Arbeit mögliche Angriffsszenarien auf das System und arbeitet heraus, dass ein Angreifer in der Lage sein muss die Blockchain schneller zu erweitern als die anderen Netzwerkteilnehmer, um Schaden anzurichten. Gleichzeitig ermittelt er die Wahrscheinlichkeit, das ein Angreifer dazu in der Lage ist [16, 11. Calculations]. Diese Garantie an die Sicherheit bringt jedoch auch einige Voraussetzungen mit sich. Die Blockchain bleibt nur so lange sicher, wie der Großteil der Rechenleistung im Netzwerk dafür eingesetzt wird neue und unverfälschte Blöcke zu erstellen. Und an genau diesem Punkt finden sich einige der größten Schwachpunkte von Proof of Work. Als erstes ist sicherlich der enorme und viel kritisierte Energieverbrauch zu nennen, der entsteht wenn große Rechnersysteme ununterbrochen mit ihrer gesamten Rechenleistung daran arbeiten neue Blöcke zu erstellen. Wer bei der Gestaltung der Blockchain mitarbeiten möchte und seinen Teil leisten will, um für ihre Sicherheit zu garantieren, der ist gezwungen große Mengen an Energie in Form von Rechenzeit zu investieren und zwar von Inbetriebnahme des

Netzwerks bis zu seiner eventuellen Abschaltung, zu der es im Optimalfall niemals kommen wird. Dieses Problem ist besonders für die zugrundeliegende Aufgabenstellung dieser Arbeit ein Problem, da sie sich mit heterogenen Geräten beschäftigt. Es besteht ein sehr realistisches Risiko, dass nicht alle verwendeten Geräte in einem Netzwerk in der Lage sein werden diesen Energieverbrauch, konstanten Betrieb und die Rechenleistung zu stemmen.
Der zweite große Kritikpunkt bezieht sich auf die Verteilung von Rechenleistung, da diese im direkten Zusammenhang mit sowohl der Sicherheit, als auch der Dezentralität des Systems steht. Dieses Problem wird oft auch als die 51%-Problematik bezeichnet. Sollte es einem Netzwerkteilnehmer oder auch einem Zusammenschluss von Netzwerkteilnehmern gelingen über die Hälfte der Rechenleistung im System auf sich zu vereinigen erlangt dieser großen Einfluss auf die weitere Entwicklung der Blockchain. Angriffe auf die Blockchain werden deutlich einfacher und das System ist nicht länger dezentral.
Eine weitere zentrale Eigenschaft von Proof of Work ist die Ausfallsicherheit. Einzelne Netzwerkteilnehmer müssen jederzeit dazu in der Lage sein das Netzwerk zu verlassen und zu betreten, ohne dabei jemals einen inkonsistenten Datenstand vorzufinden. Es geht schließlich um eine Währung. Es darf kein Geld verloren gehen und auch kein Geld dupliziert werden. Diese Eigenschaft bringt weitere Probleme mit sich. So ist die Datenspeicherung in der Blockchain häufig sehr redundant gehalten. Im Prinzip müsste jeder Netzwerkteilnehmer zu jederzeit die vollständige Blockchain lokal speichern. Um die Netzwerkteilnehmer etwas zu entlasten, wurden jedoch einige Kompromisse eingegangen. So können zum Beispiel Arten von Netzwerkteilnehmern unterschieden werden. Sogenannte Ligthweight Nodes, die nur die Blockheader lokal gespeichert haben und lediglich dazu in der Lage sind Transaktionen durchzuführen und diese zu validieren, und Full Nodes, die die gesamte Blockchain gespeichert haben und damit in der Lage sind selbst neue Blöcke zu erstellen und die Blockchain an anderen Netzwerkteilnehmer zu versenden. Während die Möglichkeit der verschiedenen Rollen von Netzwerkteilnehmern sicherlich großes Potenzial für den Einsatz von heterogenen Geräten darstellt, führt die generell hohe Redundanz der gespeicherten Daten je nach Anwendungsgebiet sicherlich zu Problemen. Dieser Punkt ist ein Argument dafür ein Verfahren wie Proof of Work in einem Anwendungsgebiet einzusetzen, welches von hoher Ausfallsicherheit profitiert und für das die hohe Redundanz einen weniger gravierenden Nachteil darstellt.
Der letzte Kritikpunkt stellt weniger ein Problem von Proof of Work als eher ein Problem der Blockchain im Allgemeinen dar. Dementsprechend wird er sich auch bei einigen der anderen Consensus Verfahren in verschieden starken Ausprägungen wiederfinden. Es geht dabei um die generell hohen Kommunikationskosten. Das dezentralisierte System Blockchain erfordert sehr viel Kommunikation zwischen den Netzwerkteilnehmern. Es müssen nicht nur neue Transaktionen an alle möglichen Blockersteller weitergeleitet werden, auch die neuen Blöcke müssen verbreitet werden. Weitere Kommunikation ist erforderlich, um die längste Blockchain im Netzwerk zu ermitteln und somit Branches aufzulösen. Zusam-

menfassend lässt sich also sagen, dass eine stete Kommunikation für die Funktion der Blockchain unerlässlich ist und dabei ebenfalls eine große Herausforderung an die Infrastruktur des Netzwerks darstellen kann.

3.2 Proof of Stake

Wie bereits zuvor in Kapitel zwei erwähnt ist Proof of Stake ein Konzept, das aus Proof of Work entstanden ist. Erklärtes Ziel war es dabei eine energiesparende Alternative zu entwickeln, die vergleichbare Sicherheit für eine Währung gewährleistet. Vorweg ist zu sagen, dass Proof of Stake sicherlich das am stärksten an eine Währung gebundene Consensus Verfahren ist. Proof of Stake benötigt explizit eine Form von Besitz, die investiert werden kann, um in der Blockchain Einfluss zu nehmen und bestimmte Operationen durchzuführen. Dies schränkt das Verfahren in seinen Anwendungsgebieten zunächst einmal ein. Sicherlich gibt es Möglichkeiten Anwendungsgebiete um ein solches Konzept von Besitz zu erweitern. Ein gutes Beispiel dafür bietet Ethereum, welches im vorhergegangenen Kapitel in Abschnitt 2.4 bereits kurz vorgestellt wurde. Hier wurden Tokens, sogenannte Ether, eingeführt, die von den Nutzern des Systems verwendet werden können, um bestimmte Operationen innerhalb der Blockchain anzustoßen. Sie fungieren also nicht als klassische Währung, sondern dienen lediglich als Zugangstokens für Funktionalitäten innerhalb des Systems selbst. Es ist jedoch fraglich, ob dies zu rechtfertigen ist, sollte Proof of Stake nicht deutlich besser abschneiden als andere Verfahren.
Proof of Stake verfolgt einen sehr einfachen und direkten Ansatz, um sein Ziel zu erreichen. Indem die Komplexität des mathematischen Problems bei Proof of Work drastisch reduziert wird, sinkt automatisch die benötigte Rechenleistung und damit auch die verbrauchte Energie. Doch stellt diese Änderung eine Gefahr für die Sicherheit der Blockchain dar? Das hängt noch stärker von den Netzwerkteilnehmern ab, als bei Proof of Work. Da die Vereinfachung des mathematischen Problems vom Besitz des jeweiligen Netzwerkteilnehmers abhängt, erhalten die Nutzer mit dem meisten Kapital den größten Einfluss über das System. Das Problem hat sich also vom Besitz von Rechenleistung auf den Besitz von Kapital innerhalb des Systems verschoben. Zusätzlich kommt hinzu, dass die Erstellung von neuen Blöcken in Form von Transaktionskosten und des Blockrewards neues Kapital in die Hände der Einflussreichen bringt. Es entwickelt sich also ein Schneeballprinzip, welches Netzwerkteilnehmern, die bereits über viel Einfluss verfügen, noch mehr Einfluss gewährt. Dieser Schneeballeffekt wird teilweise unterdrückt durch die Funktionsweise des Coin Ages, da das Investment des Coin Ages zumindest zeitweilig einen Verlust von Einfluss bedeutet, der allerdings geringfügiger wird, umso größer der Unterschied im Besitz der verschiedenen Netzwerkteilnehmer ist. Die Sicherheit des Systems lässt sich also mehr oder weniger auf die Vertrauenswürdigkeit der User mit dem Großteil des Besitzes herunterbrechen.
Sehr gelungen ist dabei, wie es Proof of Stake durch die Einführung des Coin Ages gelingt,

dass die Netzwerkteilnehmer nicht ihren tatsächlichen Besitz investieren müssen, sondern lediglich die Zeit, über die sie diesen Besitz bereits haben. Sollte sich ein Investment also einmal nicht auszahlen, verlieren sie notwendigerweise Einfluss auf die Blockchain, aber nicht ihren tatsächlichen Besitz.
Doch Proof of Stake wirft neben den zuvor bereits behandelten Themen auch ganz neue Fragen auf. Möchte man ein solches System implementieren so gilt es zum Beispiel eine Strategie zu entwickeln, nach welcher die Miner ihr Kapital investieren, um sich die Erstellung von Blöcken zu erleichtern. Auch hier verbirgt sich gerade im Zusammenhang mit heterogenen Geräten ein großes Potenzial. So könnten die unterschiedlichen Geräte sehr unterschiedlich Strategien verwenden. Geräte mit nur sehr geringer Rechenleistung beispielsweise könnten seltenere und dafür höhere Investments machen, um ihre geringe Rechenleistung optimal zu nutzen.
Rein konzeptionell sollte Proof of Stake weder bei den hohen Kommunikationskosten der Blockchain noch bei dem hohen Speicheraufwand von Proof of Work einen Vorteil herausarbeiten können. Näheres dazu wird sich hoffentlich in den Ergebnissen der folgenden Experimente zeigen.

3.3 Practical Proof of Kernel Work

Xain verfolgt mit ihrem Practical Proof of Kernel Work einen anderen Ansatz zur Verbesserung von Proof of Work. Da Xains Einsatz der Blockchain nicht an eine Kryptowährung gebunden ist, ist auch Practical Proof of Kernel Work deutlich vielseitiger einsetzbar. Mit Practical Proof of Kernel Work ist es gelungen die meisten vorteilhaften Eigenschaften von Proof of Work zu erhalten und gleichzeitig die größten Probleme zu beheben.
Zunächst einmal ist der Energieverbrauch deutlich verringert, da der Zugang zum eigentlichen Proof of Work durch die Whitelist, die dynamischen Regeln und die Zufallsauswahl stark eingeschränkt wird. Dies führt dazu, dass zu jedem Zeitpunkt deutlich weniger Netzwerkteilnehmer am Proof of Work arbeiten. Gleichzeitig hilft diese Form der Zugangsbeschränkung auch dabei die 51%-Problematik in den Griff zu bekommen, da weder große Mengen an Rechenleistung, noch große Mengen an Kapital die Wahrscheinlichkeit erhöhen für die Erstellung des nächsten Blocks selektiert zu werden. Dadurch lässt sich bei großen Mengen an Netzwerkteilnehmern nahezu ausschließen, dass neue Blöcke immer von denselben Netzwerkteilnehmern erstellt werden, und diese damit besonders großen Einfluss über die Blockchain gewinnen. Dies wird insbesondere durch die dynamischen Regeln unterstützt, da eine solche Regel zum Beispiel lauten könnte: „Wer einen Block erstellt hat, der ist für die nächsten drei Blöcke vom Auswahlverfahren ausgeschlossen! “.
Große Vorteile bringt dabei auch die Art und Weise, auf welche die Zufallsauswahl durchgeführt wird. Verifiable Random Functions erlauben es zufällig Netzwerkteilnehmer auszuwählen, ohne das andere Netzwerkteilnehmer wissen können, wer ausgewählt wurde.

Dies macht es für Angreifer unmöglich gezielt Netzwerkteilnehmer zu kompromittieren, die zukünftige Blöcke erstellen werden. Gleichzeitig können ausgewählte Netzwerkteilnehmer durch das versenden ihres Proofs jederzeit beweisen, dass sie tatsächlich ausgewählt wurden.
Practical Proof of Kernel Work hilft auch dabei die Kommunikationskosten im Netzwerk zu verringern. Durch die geringere Anzahl an Netzwerkteilnehmern, die zeitgleich das Proof of Work durchführen, sinkt die Wahrscheinlichkeit, dass Branches der Blockchain entstehen. Außerdem besteht die Möglichkeit Transaktionen gezielter an Netzwerkteilnehmer zu senden, die zukünftige Blöcke erstellen werden. Transaktionen müssen also nicht mehr immer an alle Netzwerkteilnehmer gesendet werden.
Natürlich bringt auch Practical Proof of Kernel Work einige Nachteile mit sich. Die Speicherung einer Whitelist auf der Blockchain ist ein potenzielles Problem für die Skalierbarkeit des Netzwerks. In großen Netzwerken würde eine solche Liste von Netzwerkteilnehmern viel Speicherplatz verbrauchen, den die leistungsschwächeren heterogenen Geräte eventuell nicht aufbringen können. Je nach System würde es eventuell Sinn machen stattdessen eine Blacklist zu verwenden, falls die Anzahl der vertrauenswürdigen Netzwerkteilnehmer die der nicht vertrauenswürdigen deutlich überwiegt. Auch die Durchführung einer Verifiable Random Function stellt je nach Anforderungen an die Sicherheit ebendieser eine Herausforderung für die heterogenen Geräte dar.

3.4 Distributed Proof of Work

Distributed Proof of Work nutzt einen sehr ähnlichen Ansatz wie Practical Proof of Kernel Work. Auch hier sollen die Probleme von Proof of Work behoben werden, indem der Zugang zum eigentlichen Proof of Work beschränkt wird. Distributed Proof of Work verwendet hier wie in Kapitel zwei beschrieben jedoch deutlich einfachere Methoden.
Es wird sowohl auf die Whitelist, als auch auf die dynamischen Regeln verzichtet. Die Sicherheit des Gesamtsystems wird also leicht zurückgestellt, um das Verfahren im Gegenzug einfacher zu halten. Dies ist jedoch gerade für die heterogenen Geräte ein großer Unterschied, da die Speicherproblematik durch die Whitelist wegfällt, und geringere Komplexität generell einen Vorteil für die leistungsschwächeren Geräte darstellt. Generell summieren sich die rechnerischen Anforderungen aller Prüfungen, die in das Mining neuer Blöcke integriert sind, schnell auf, da es sich um eine fundamentale Funktion der Blockchain handelt, die notwendig ist, um das ganze System am laufen zu halten. Dementsprechend häufig werden sie auch ausgeführt und verursachen damit eine entsprechend hohe Belastung.
Ein weiteres Problem von Distributed Proof of Work sind die relativ ungenauen Beschreibungen des Verfahrens durch die Autoren [3]. Ein Beispiel dafür ist die Beschreibung des zufälligen Auswahlprozesses von Blockerstellern. Auch die starke Einbindung in die politische Agenda von Cicada hat negative Auswirkungen auf das Verfahren. So ist das ursprüngliche

Konzept von Distributed Proof of Work sehr stark an einen konkreten Anwendungsfall gebunden. Dieser Anwendungsfall ist ein System zur Umsetzung einer direkten Demokratie und dementsprechend mit einigen Restriktionen verbunden. Eine solche Restriktion ist das eins zu eins Verhältnis von Netzwerkteilnehmern zu Menschen, durch die Verwendung eines Human Unique Identifiers. Um dem Verfahren wirklich gerecht werden zu können und einen fairen Vergleich zu den anderen Consensus Verfahren ziehen zu können, ist es notwendig das Verfahren auf seine technischen Grundsätze zu reduzieren und von der politischen Motivation losgelöst zu betrachten. Die grundsätzliche Idee ist aber sehr vielversprechend und sollte großes Potenzial für die Verwendung mit einer Blockchain in einem Netzwerk heterogener Geräte bieten.

Kapitel 4

Implementierung der Consensus Verfahren

4.1 Vorwort zur Implementierung

Da gerade bei den beiden neueren und weniger verbreiteten Consensus Verfahren Distributed Proof of Work und Practical Proof of Kernel Work kaum/keine fertigen Implementierungen existieren beziehungsweise diese nicht frei zugänglich sind, war es leider notwendig diese im Rahmen dieser Arbeit selbst zu implementieren. Um die Vergleichbarkeit der Implementierungen zu gewährleisten, wurden dementsprechend auch Proof of Work und Proof of Stake für diese Arbeit implementiert. Als Grundlage für die Implementierung diente dabei ein Tutorial von HackerNoon[4]. Alle Implementierungen stehen Online[1] zur freien Verfügung.
Es folgt die Beschreibung der Implementierung der verschiedenen Consensus Verfahren. Dabei soll es vor allem um die getroffenen Designentscheidungen und ihre Begründungen sowie aufgetretene Probleme und Herausforderungen bei der Implementierung gehen.
Als Programmiersprache für das Projekt wurde Python gewählt. Dies hat mehrere Gründe. Zunächst einmal soll diese Wahl ermöglichen in Zukunft einen Wechsel auf MircoPython zu erleichtern, sobald dieses über einen größere Codebasis verfügt. Dieser mögliche Wechsel bietet sich an, da MicroPython eine Variante von Python ist, die explizit für die Verwendung auf leistungsschwachen Geräten entwickelt wurde. Das macht sie ideal, um sie auf heterogenen Geräten einzusetzen. Außerdem erfreut sich Python großer Beliebtheit und Verbreitung, sodass es über eine entsprechend große Codebasis an unterstützenden Paketen und Funktionalitäten verfügt.
Generell war es das Ziel ein Programm zu entwickeln, welches in beliebig vielen Instanzen ausgeführt werden kann. Eine Instanz repräsentiert dabei einen Netzwerkteilnehmer. Die

[1] `https://bitbucket.org/cedric_sanders/blockchain-experiments/src/master/`

Netzwerkteilnehmer kommunizieren dabei über http-Requests untereinander und müssen in der Lage sein folgende Operationen durchzuführen:

1. Durchführen von Transaktionen
2. Beteiligung an der Erstellung neuer Blöcke
3. Synchronisation der Blockchain
4. Input von Daten von außerhalb (Über eine Datei)
5. Messung von Statistiken für die Evaluation
6. Ausführung eines Verfahrens zur Datenanalyse

Zu den wichtigsten Zielen gehörte es dabei die verschiedenen Verfahren möglichst vergleichbar und einfach zu halten. Die Implementierung sollten möglichst verständlich sein und dabei den Fokus auf die Kernkonzepte und Besonderheiten der einzelnen Verfahren legen und sich nicht in der Umsetzung von Spezialfällen verlieren. Am Ende des Kapitels ist auch zu jedem Verfahren der jeweilige Mining Prozess kurz als Pseudocode dargestellt, da dieser den zentralen Unterschied zwischen den Verfahren repräsentiert. Für einen detaillierteren Einblick in die Implementierungen steht die bereits erwähnte Online Version zur Verfügung.

4.2 Proof of Work

Proof of Work ist nicht nur das einfachste, der hier behandelten Verfahren, sondern auch das am besten dokumentierte. Dementsprechend war die Implementierung alles in allem unproblematisch. Genau wie die drei folgenden Verfahren ist das Programm in drei Teile unterteilt. Einen Webserver, der für die Kommunikation zwischen den Netzwerkteilnehmern zuständig ist, einen separaten Programmteil für das Mining, der alles nötige abhandelt, um erhaltene Transaktionen zu neuen Blöcken zusammenzufügen, und einen letzten Teil zum Empfang und zur Analyse von Daten. Durch die relativ simple und klar definierte Struktur von Proof of Work gab es hier keine nennenswerten Designentscheidungen zu treffen. Bei der Auswahl der Mining Difficulty fiel die Wahl auf eine Menge von sechs aufeinanderfolgenden Nullen. Durch die Durchführung einiger kurzer Tests hat sich gezeigt, dass diese Konfiguration in den meisten Fällen zu einer durchschnittlichen Mining Zeit von um die vierzig Sekunden geführt hat. Dies harmonisiert also sehr gut mit dem einlesen neuer Sensordaten, welches ebenfalls in Abständen von in etwas vierzig Sekunden stattfindet.[2]

[2]Siehe Algorithmus 4.1 in Abschnitt 4.6.

4.3 Proof of Stake

Die Implementierung von Proof of Stake ist deutlich interessanter als die von Proof of Work. Auch sie weist die selbe bereits beschriebene Grundstruktur auf. Während zwei der drei Programmteile im Prinzip von Proof of Work übernommen werden konnten, so unterscheidet sich der dritte Programmteil zum Mining doch deutlich.
Zunächst einmal war es notwendig einige grundsätzliche Änderungen an der Datenstruktur der Blockchain vorzunehmen, um das benötigte Coin Age verwenden zu können. Zu diesem Zweck mussten Funktionalitäten geschaffen werden, die es erlauben zu jedem Zeitpunkt das Coin Age zu bestimmen, über das ein Netzwerkteilnehmer verfügt und ausgegebenes Coin Age korrekt abzuziehen. Außerdem musste ein neuer Transaktionstyp eingeführt werden, der das getätigte Investment symbolisiert, welches Netzwerkteilnehmer vornehmen, um das Mining zukünftiger Blöcke zu erleichtern.
Als wichtige Designentscheidung galt es die in Abschnitt 3.2 thematisierte Investmentstrategie für die verschiedenen Netzwerkteilnehmer festzulegen. Aufgrund der hohen Komplexität dieses Themas und, um die verschiedenen Implementationen möglichst einfach und unspezifisch[3] zu halten, fiel die Entscheidung hier auf zufallsbedingte Investments. Jeder Netzwerkteilnehmer investiert also einen zufällige Menge, des ihm zur Verfügung stehenden Coin Ages. Es gilt natürlich auch hier festzulegen, inwiefern das investierte Coin Age sich auf die Mining Difficulty auswirkt. Generell ist zu sagen, dass in dem gegebenen System kaum Währungseinheiten in den Umlauf gebracht werden. Lediglich die Erstellung eines neuen Blocks führt durch den Block Reward zur Ausschüttung von Währungseinheiten. Das Coin Age ist damit im Bezug auf sein Wachstum sehr limitiert und sollte dementsprechend große Auswirkungen auf die Mining Difficulty haben, um die Unterschiede zum klassischen Proof of Work besser herausstellen zu können. Eine Alternative ist natürlich eine Initialisierung der Netzwerkteilnehmer mit einer gewissen Menge an Währungseinheiten. Die Wahl fiel letztendlich auf folgende Formel: $max(0, 6 - n/100)$, wobei n der Anzahl an investiertem Coin Age entspricht und $n/100$ offensichtlich ganzzahlig gerundet wird.[4]

4.4 Practical Proof of Kernel Work

Practical Proof of Kernel Work macht in der Komplexität im Vergleich zu den bisherigen Implementierungen nochmal einen großen Sprung. Dieser bezieht sich erneut primär auf den Teil des Programms, welcher für das Mining verantwortlich ist. In diesem Fall sind aber auch einige Erweiterungen des Webservers notwendig. Durch die große Synchronisierung des Mining Vorgangs im Rahmen der in Practical Proof of Kernel Work eingeführten Mining Races ist eine ganze Reihe neuer Kommunikationsvorgänge notwendig, um den rei-

[3] Dadurch soll das Ergebnis allgemeingültiger bleiben.
[4] Siehe Algorithmus 4.2 in Abschnitt 4.6.

bungslosen Ablauf zu gewährleisten.
Eine der größten Erweiterung ist sicherlich die Verwendung von Verifiable Random Functions, die auch direkt mit einer Designentscheidung verbunden ist, da es doch sehr unterschiedliche Ansätze zur Verwendung ebendieser gibt. Die Entscheidung fiel auf den in [5, Definition 4.1] beschriebenen Ansatz. Ein großes Problem stellte dabei lange Zeit die Synchronisierung der für die Verifiable Random Functions benötigten Seeds dar. Durch leichte Laufzeit Unterschiede zwischen den Netzwerkteilnehmern kam es immer wieder zu leichten Verschiebungen zwischen den Seeds, sodass eine gewisse Toleranz bei der Überprüfung dieser eingeführt wurde. Es wird also nicht nur der aktuelle, sondern auch der Seed des vorherigen bzw. nachfolgenden Mining Race akzeptiert. Ein Mining Race kann beginnen, sobald eine ausreichend große Anzahl an Netzwerkteilnehmern für dieses selektiert wurde. Diese Anzahl stellt einen interessanten Steuerungsparameter des Systems dar. Über ihn kann wie auch mit der Mining Difficulty die Geschwindigkeit der Mining Races konfiguriert werden. Zeitgleich bestimmt er aber auch wie viele Netzwerkteilnehmer gleichzeitig am Mining beteiligt sind und damit große Mengen an Rechenleistung und Strom verbrauchen. Innerhalb dieser Implementierung fiel die Wahl auf eine Menge von fünfzig Prozent der Netzwerkteilnehmer. Dies dient vor allem dazu sicher zu stellen, dass die Erstellung neuer Blöcke nicht zu lange dauert und damit der Datendurchsatz sinkt. In sehr großen Netzwerken ist es vermutlich ohne Probleme möglich diesen Parameter zu reduzieren.
Eine wichtige Designentscheidung im Bezug auf die Verifiable Random Functions ist dabei, wann das Ergebnis der Lotterie bekannt gegeben wird. Gibt man das Ergebnis noch vor der Erstellung des Blocks bekannt, so können die anderen Netzwerkteilnehmer ihre Transaktionen gezielt nur an solche Knoten senden, die für die Erstellung des nächsten Blocks ausgewählt wurden, und somit Kommunikation einsparen. Gibt man das Ergebnis allerdings nicht bekannt, erhöht sich die Sicherheit des Systems leicht, da es Angreifern erschwert wird gezielt zukünftige Blockersteller zu kompromittieren. Da die Performanz der Consensus Verfahren innerhalb dieser Arbeit im Mittelpunkt steht, fiel die Entscheidung auf ersteres.
Weiterhin wurde eine Whitelist in die Blockchain integriert, welche auch durch geringfügige Änderungen durch eine Blacklist ersetzt werden kann. Als zusätzliche Regel wurde implementiert, dass der Ersteller eines Blocks für den nächsten Block vom Mining ausgeschlossen wird. Eine weitere wichtige Frage, war die Konsequenz für einen Betrugsversuch bei der Auswahl für ein Mining Race. Sollte ein Knoten, ohne dafür ausgewählt worden zu sein, einen neuen Block erstellen und diesen an andere Netzwerkteilnehmer versenden, so muss es dafür Konsequenzen geben, damit nicht eventuell ein ungültiger Block in die Blockchain integriert wird. Ein Knoten, der sich eines derartigen Betrugs schuldig macht, wird von der Whitelist gestrichen und darf damit nie wieder neue Blöcke erstellen, bis er aus irgendeinem Grund wieder der Whitelist hinzugefügt wird. Die Konsequenzen sind also

durchaus gravierend für den einzelnen Netzwerkteilnehmer, stellen aber keine Bedrohung für die Funktionalität der Blokchain dar und lassen sich sehr einfach umsetzten.[5]

4.5 Distributed Proof of Work

Distributed Proof of Work als eine weniger komplexe Variante von Practical Proof of Kernel Work deckt sich in großen Teilen im Bezug auf aufgetretene Probleme und getroffene Entscheidungen. Auch hier spielen die Verifiable Random Functions und die zusätzlichen Kommunikationsoperationen zur Verwaltung der Mining Races sicherlich die zentrale Rolle. Während auf die Whitelist sowie die zusätzlichen Regeln entsprechend verzichtet werden konnte. Wie auch bei Pracitcal Proof of Kernel Work müssen zum Beginn eines Mining Race fünfzig Prozent der Netzwerkteilnehmer ausgewählt worden sein. Die Ahndung von Betrugsversuchen bei der Erstellung neuer Blöcke ist bei Distributed Proof of Work weniger gravierend. Da hier keine Whitelist existiert, von der der Netzwerkteilnehmer ausgeschlossen werden könnte, wird er lediglich vom aktuellen Mining Race ausgeschlossen und darf im nächsten Mining Race, falls selektiert, wieder teilnehmen. Das Risiko, dass ein solcher Betrug gelingt, ist sehr geringfügig, und daher ist nicht mit Problemen zu rechnen, sollte der Knoten wieder versuchen zu betrügen. Auch bei Practical Proof of Kernel Work würde es vermutlich genügen den Netzwerkteilnehmer vom aktuellen Mining Race auszuschließen, die Entfernung von der Whitelist spart jedoch im weiteren Verlauf des Programms Rechenleistung und Speicherplatz.[6]

4.6 Pseudocode

1: $nonce \leftarrow 0$
2: **while** *proof* is not valid **do**
3: $\quad$ *proof* $\leftarrow$ *ProofofWork(Last Block, nonce)*
4: $\quad$ $nonce \leftarrow nonce + 1$
5: **end while**
6: *RewardMiner()*
7: *CreateBlock(Last Block, proof)*

Algorithmus 4.1: Mining with Proof of Work

[5] Siehe Algorithmus 4.4 in Abschnitt 4.6.
[6] Siehe Algorithmus 4.3 in Abschnitt 4.6.

```
1: coinage ← CalculateCoinAge()
2: investment ← Random(coinage)
3: MakeInvestment()
4: nonce ← 0
5: while proof is not valid do
6:    proof ← ProofofWork(Last Block, nonce, investment)
7:    nonce ← nonce + 1
8: end while
9: RewardMiner()
10: CreateBlock(Last Block, proof)
```

Algorithmus 4.2: Mining with Proof of Stake

```
1: selected ← False
2: if enough nodes selected then
3:    if selected then
4:       nonce ← 0
5:       while proof is not valid do
6:          proof ← ProofofWork(Last Block, nonce)
7:          nonce ← nonce + 1
8:       end while
9:       RewardMiner()
10:      CreateBlock(Last Block, proof)
11:   else
12:      Wait for next mining race
13:   end if
14: else
15:   selected ← VerifiableRandomFunction(seed, difficulty)
16:   if selected then
17:      Let the other nodes verify the Verifiable Random Function
18:   else
19:      Wait for other nodes to do the lottery
20:      if not enough nodes selected then
21:         Reduce the difficulty of being selected
22:      end if
23:   end if
24: end if
```

Algorithmus 4.3: Mining with Distributed Proof of Work

```
1: selected ← False
2: if enough nodes selected then
3:   if selected then
4:     nonce ← 0
5:     while proof is not valid do
6:       proof ← ProofofWork(Last Block, nonce)
7:       nonce ← nonce + 1
8:     end while
9:     RewardMiner()
10:    CreateBlock(Last Block, proof)
11:  else
12:    Wait for next mining race
13:  end if
14: else
15:  if Node on Whitelist then
16:    if CheckRuleset() then
17:      if VerifiableRandomFunction(seed, difficulty) then
18:        selected ← True
19:      end if
20:    end if
21:  end if
22:  if selected then
23:    Let the other nodes verify the Selection
24:  else
25:    Wait for other nodes to do the lottery
26:    if not enough nodes selected then
27:      Reduce the difficulty of being selected
28:    end if
29:  end if
30: end if
```

Algorithmus 4.4: Mining with Practical Proof of Kernel Work

Kapitel 5

Experimente und Evaluation

5.1 Anwendungsfall

Um ein abschließendes Urteil über die Eignung von Blockchains zur Analyse und Verwaltung von Daten in dezentralen Netzwerken heterogener Geräte fällen zu können, ist es natürlich notwendig ein entsprechendes Datenanalyseverfahren in einem solchen Kontext zu testen. Da der Fokus dieser Arbeit aber zunächst einmal auf der generellen Eignung von Blockchains und nicht dem spezifischeren Wechselspiel mit bestimmten Datenanalyseverfahren liegt, bietet es sich an ein Verfahren mit entsprechend günstigen Eigenschaften zu wählen.
Ein solches Verfahren ist Geometric Monitoring [18]. Es wurde explizit zur Dezentralisierung von Datenanalysen entwickelt und beruht darüber hinaus auf einem relativ simplen Grundkonzept. Diese Eigenschaften ermöglichen es Geometric Monitoring ohne große Komplikationen in eine Blockchain zu integrieren.
Als Anwendungsfall bietet sich die Überwachung von Grenzwerten in openSenseMap[1] an. Dabei handelt es sich um eine Plattform für offene Sensordaten, welche vom Institut für Geoinformatik in Münster ins Leben gerufen wurde. Jeder kann dort Sensoren registrieren und so zur Erweiterung des Netzwerks beitragen. Die gemessenen Sensordaten stehen dabei unter der „Public Domain Dedication and License 1.0“[2] und können von jedem frei verwendet werden. Die Sensoren von openSenseMap liefern die unterschiedlichsten Daten, wie zum Beispiel Windgeschwindigkeiten, Temperaturen und Schadstoffwerte. Die Werte solcher Sensoren können nun also als Input für die Netzwerkteilnehmer der Blockchain dienen, auf welchen dann Geometric Monitoring durchgeführt wird.

[1] `https://opensensemap.org/`
[2] `https://opendatacommons.org/licenses/pddl/summary/`

5.1.1 Geometric Monitoring

Geometric Monitoring ist ein Konzept, welches erstmals 2007 in [18] behandelt wurde. Es beschäftigt sich damit, wie in einem Netzwerk bei der Überwachung einer globalen Funktion die Kommunikationskosten reduziert werden können. Besonders interessant für den genannten Anwendungsfall sind auch [10] und [8], da sich diese Arbeiten mit der Verwendung von Geometric Monitoring auf Streams befassen und somit sehr geeignet für die Anwendung auf die Sensoren sind.
Geometric Monitoring liegt das Problem zu Grunde, dass das Überprüfen einer globalen Funktion in einem zentralisierten Netzwerk sehr hohe Kommunikation voraussetzt, da alle Daten an einer zentralen Stelle gesammelt werden müssen, um die Funktion zu berechnen. Die Lösungsidee von Geometric Monitoring besteht darin die Häufigkeit der Berechnung dieser globalen Funktion zu begrenzen, indem jeder Netzwerkteilnehmer zunächst eine lokale Bedingung überprüft, die verletzt werden muss, damit die globale Funktion berechnet wird. Diese lokalen Bedingungen müssen dabei drei Eigenschaften aufweisen:

5.1 Definition (Lokale Bedingungen). vgl. [10, Definition 1.1]
Gegeben ein Netzwerk mit Netzwerkteilnehmern $N_1, ..., N_k$, wobei jedem Netzwerkteilnehmer N_i ein Wert v_i zugeordnet wird. Sei f eine Funktion und T ein globaler Grenzwert, dann sind die lokalen Bedingungen so zu wählen, dass folgendes gilt:

1. Korrektheit: Solange alle lokalen Bedingungen gelten, gilt auch $f(v_1, ..., v_k) \leq T$.
2. Kommunikationseffizienz: Die Anzahl an Überschreitungen der lokalen Bedingungen ist minimal.
3. Effiziente Berechenbarkeit: Die Komplexität der Überprüfung der lokalen Bedingungen ist minimal.

Solche lokalen Bedingungen zu finden ist dabei dank der in [10] beschriebenen Methode convex bound erstaunlich einfach, wenn f eine konvexe Funktion ist. Es gilt lediglich eine konvexe Funktion c zu finden, die einfach zu berechnen ist und eine enge obere Grenze für f bildet.

Im Zusammenhang mit der Blockchain ergibt sich sogar eine völlig neue Möglichkeit für das Geometric Monitoring. Während normalerweise einige Netzwerkteilnehmer als sogenannte Koordinatoren fungieren müssen, um im Falle einer Grenzwertüberschreitung die Daten zur Berechnung der globalen Funktion von den anderen Netzwerkteilnehmern zu sammeln, ist dies im Kontext der Blockchain nicht notwendig. Jeder Knoten kann als Koordinator fungieren, da er über die Blockchain Zugriff auf alle bereits übertragenen Daten hat. Hier kann der ursprüngliche Mechanismus also vereinfacht, und Kommunikation eingespart werden.

5.2 Versuchsaufbau

Wie bereits in Abschnitt 5.1 beschrieben werden als Testdaten Daten von openSenseMap[3] verwendet. openSenseMap stellt insgesamt Werte von 3909 sogenannten senseBoxen zur Verfügung, die grundsätzlich über die ganze Welt verteilt sind, aber primär in Europa und im Speziellen in Deutschland liegen. Die Daten, die von den verschiedenen Sensoren gemessen werden, sind dabei sehr unterschiedlich. Während es zu sehr speziellen Daten wie in etwa Radioaktivität nur einige wenige Sensoren gibt, so sind Daten wie die Temperatur und Windgeschwindigkeit von fast allen Sensoren messbar. Um Geometric Monitoring zu testen, fiel die Wahl aus mehreren Gründen auf die Temperatur. Zu dieser lassen sich auf der openSenseMap die Daten vieler Sensoren ablesen. Außerdem lässt sich hier leicht ein entsprechender Grenzwert festlegen und etwaige Ergebnisse sind leicht und intuitiv ohne viel weiteres Wissen interpretierbar.

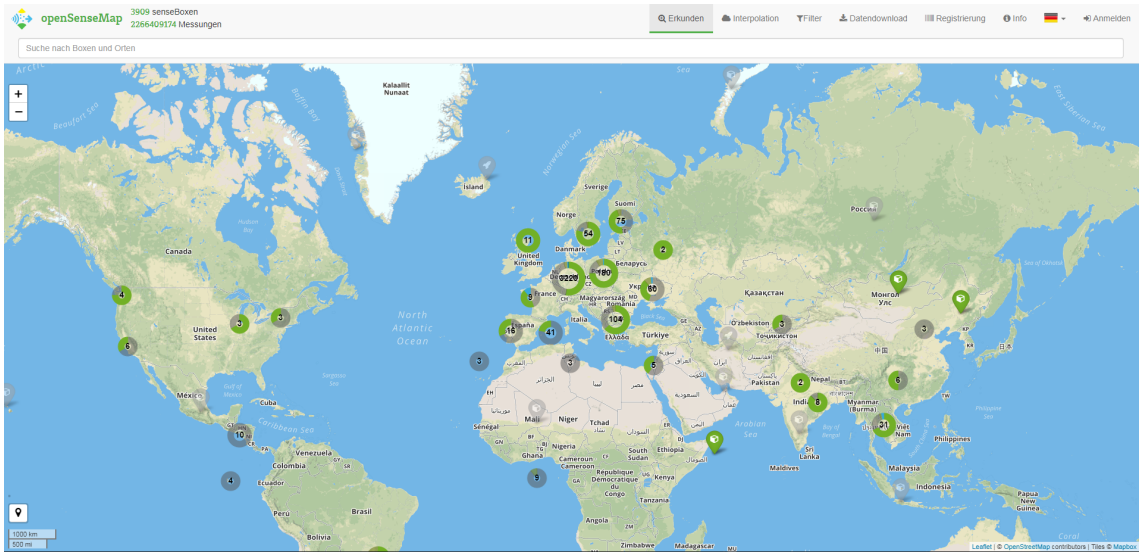

Abbildung 5.1: Globale Verteilung der Sensoren von openSenseMap [6]

Verwendet wird also eine csv-Datei mit den Messergebnissen von 1198 Sensoren über eine Zeitspanne vom 23.03.2019 bis zum 24.03.2019, die sich räumlich in Deutschland[4] befinden.

Zum Starten der Experimente wird ein zweigeteiltes Python-Skript verwendet. Der erste Teil des Skripts initialisiert dabei das Blockchain-Netzwerk und erhält vier verschiedene Parameter. Der erste ist die Anzahl an Netzwerkteilnehmern, die simuliert werden sollen. Als nächstes wird das Python-Skript angegeben, welches die Implementierung des

[3] `https://opensensemap.org/`
[4] Koordinaten als bbox 5.98865807458, 47.3024876979, 15.0169958839, 54.983104153

gewünschten Consensus Verfahrens enthält. Der dritte Parameter benennt die csv-Datei, welche die Daten von openSenseMap enthält. Über den letzten Parameter kann schließlich festgelegt werden, ob alle Ereignisse im Netzwerk protokolliert werden sollen und, ob Statistiken für diesen Testdurchlauf erhoben werden sollen. Das Skript wird zunächst die csv-Datei in entsprechende Dateien für jeden Netzwerkteilnehmer aufteilen und dann für jeden Netzwerkteilnehmer einen subprocess starten, welcher einen Webserver auf localhost startet.
Der zweite Teil des Skripts stößt schließlich die Prozesse der Blockchain an. Er macht die verschiedenen Netzwerkteilnehmer zunächst einmal untereinander bekannt und startet dann bei jedem von ihnen den Mining Prozess und das Einlesen der Daten aus der jeweiligen Sensordatei. Außerdem ist er dafür verantwortlich in regelmäßigen Abständen die aktuellen Statistiken der einzelnen Netzwerkteilnehmer anzufragen. Die Laufzeit kann als eine Anzahl von zehn Minuten Intervallen angegeben werden.
Die Experimente werden zunächst mit einer kleinen Anzahl an Netzwerkteilnehmern starten und diese dann nach und nach erhöhen, bis die Rechenleistung, der Speicher oder die Anzahl an Sensordaten nicht mehr genügt, um weitere Netzwerkteilnehmer zu simulieren. Ein Experiment soll dabei jeweils eine Stunde dauern.
Anschließend gilt es die generierten Logfiles nachzubearbeiten und die ermittelten Kennzahlen graphisch aufzuarbeiten.

Die Experimente wurden auf dem Rechnercluster des Lehrstuhls für Künstliche Intelligenz(LS VIII) der Technischen Universiät Dortmund durchgeführt. Eine Schritt für Schritt Anleitung zur Durchführung von Experimenten findet sich im Anhang.

5.3 Ergebnisse

Die Experimente wurden für 5, 10, 20 und 40 Netzwerkteilnehmer durchgeführt. Bei 40 Netzwerkteilnehmern war die CPU-Kapazität der zur Verfügung stehenden Rechner entsprechend ausgeschöpft. Um in Zukunft noch größere Experimente durchzuführen, sollten diese direkt auf einem Sensornetzwerk ausgeführt werden.
Die Ergebnisse lassen sich in die vier Kategorien Kommunikation, Mining, Speicher und Rechenleistung unterteilen.

5.3.1 Kommunikation

Bei der Kommunikation im Netzwerk wurden die vier zentralen Kommunikationsvorgänge zwischen den Netzwerkteilnehmern gemessen. Dabei handelt es sich um das Anfragen der aktuellen Blockchain, den Erhalt von Transaktionen, den Erhalt von Blöcken und Operationen zur Koordination des Minings. Letztere treten nur bei Practical Proof of Kernel

Work und Distributed Proof of Work auf. Auch mögliche Verbindungsfehler während der Kommunikation werden hier aufgeführt.

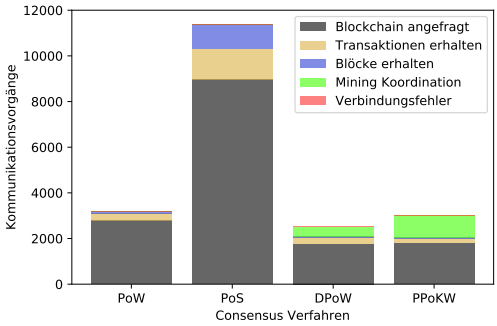

Abbildung 5.2: Durchschnittliche Kommunikationskosten bei 5 Netzwerkteilnehmern über 1 Stunde

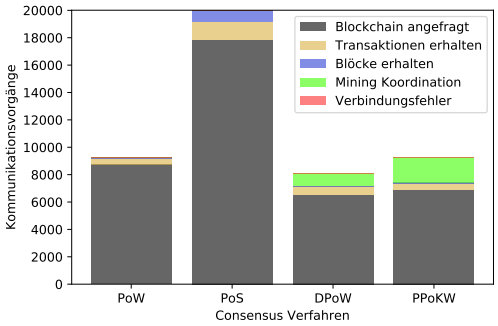

Abbildung 5.3: Durchschnittliche Kommunikationskosten bei 10 Netzwerkteilnehmern über 1 Stunde

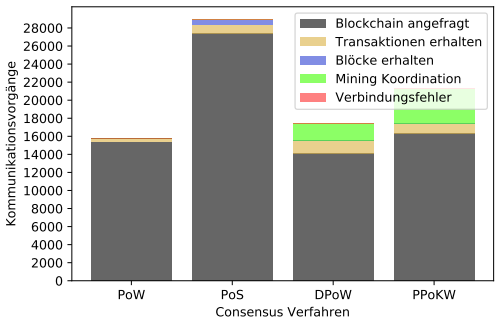

Abbildung 5.4: Durchschnittliche Kommunikationskosten bei 20 Netzwerkteilnehmern über 1 Stunde

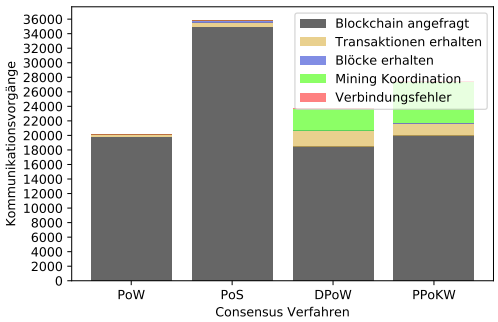

Abbildung 5.5: Durchschnittliche Kommunikationskosten bei 40 Netzwerkteilnehmern über 1 Stunde

Aus den Abbildungen 5.2 bis 5.5 lässt sich schnell erkennen, dass ein Verfahren besonders hervorsticht. Proof of Stake benötigt deutlich mehr Kommunikation, als die anderen Verfahren. Dies hat zwei Gründe. Zunächst einmal verwendet Proof of Stake Transaktionen, um die Investments von Coin Age zu kommunizieren. Da sowohl das Erstellen von Transaktionen, als auch die Abfrage des aktuellen Coin Ages einen Abgleich der Blockchain erfordern, lässt sich die vergleichsweise hohe Anzahl an Kommunikationsvorgänge dadurch erklären. Der zweite Grund wird im folgenden Abschnitt Mining klarer und hat mit der Anzahl an neu erstellten Blöcken zu tun. Generell wird klar, dass die Anfrage von Blockchains den größten Teil der Kommunikation ausmacht. Zu vernachlässigen ist der Erhalt neuer Blöcke und das Kommunizieren der Transaktionen. Was ebenfalls auffällt, sind die ähnlichen Kommunikationsvorgänge der drei anderen Consensus Verfahren, die bei steigender Anzahl von Netzwerkteilnehmern langsam auseinander driften. Um die Kommunikation deutlich zu reduzieren, muss also ein Weg gefunden werden den Bedarf an Abgleichen der Blockchain zu verringern. Zusammenfassend lässt sich damit sagen, dass

die Kommunikationskosten einen großen limitierenden Faktor darstellen und ein Problem für die Skalierbarkeit sind.

5.3.2 Mining

Bei der Kategorie Mining ging es darum die zentralen Kennzahlen einer Blockchain zu messen, also wie viele Blöcke die einzelnen Netzwerkteilnehmer im Schnitt erstellen, wie lange sie für die Erstellung eines Blocks benötigen, wie lang die Blockchain ist und wie groß die zeitlichen Abstände zwischen den einzelnen Blöcken der Blockchain sind. Diese Kennzahlen sind sehr wichtig, wenn es um den Datendurchsatz der Blockchain geht. Wenn nur einmal in der Minute ein Block erstellt wird, können auch nur dann Daten in die Blockchain übernommen werden. Benötigt ein bestimmter Anwendungsfall also einen hohen Datendurchsatz, so muss auch die Anzahl der erstellten Blöcke entsprechend groß sein. Gleichzeitig stellen diese Faktoren auch ein wichtiges Kriterium für die Sicherheit der Blockchain dar.

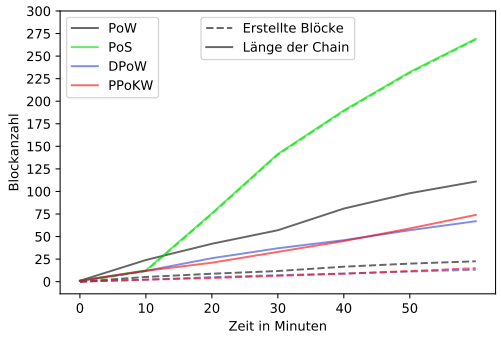

Abbildung 5.6: Durchschnittlich erstellte/totale Blöcke bei 5 Netzwerkteilnehmern über 1 Stunde

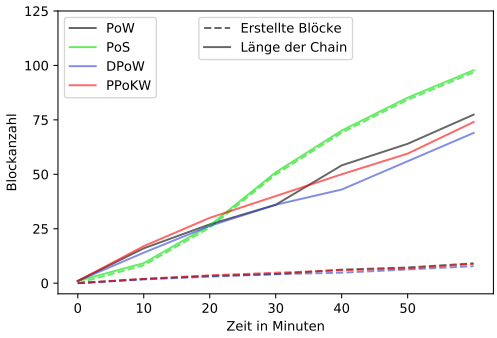

Abbildung 5.7: Durchschnittlich erstellte/totale Blöcke bei 10 Netzwerkteilnehmern über 1 Stunde

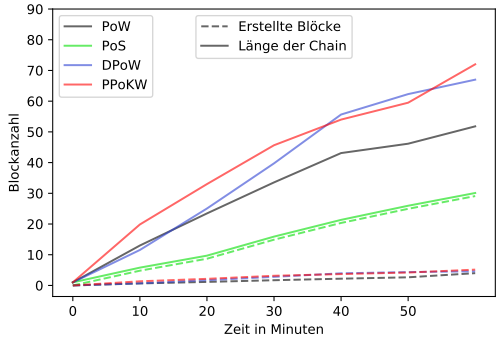

Abbildung 5.8: Durchschnittlich erstellte/totale Blöcke bei 20 Netzwerkteilnehmern über 1 Stunde

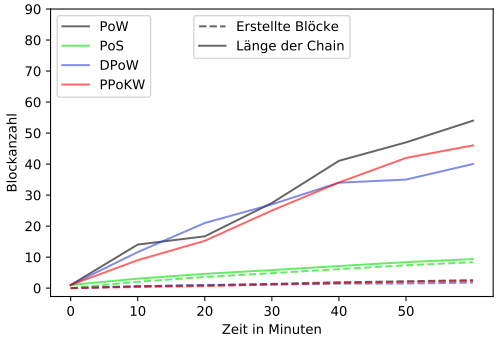

Abbildung 5.9: Durchschnittlich erstellte/totale Blöcke bei 40 Netzwerkteilnehmern über 1 Stunde

Betrachtet man nun die Abbildungen 5.6 bis 5.9 so sticht Proof of Stake erneut hervor. Dies ist aber keinesfalls verwunderlich. Hier lässt sich sehr gut der während der Analyse in Abschnitt 3.2 beschriebene Schneeballeffekt erkennen. Durch den Erhalt von Währungseinheiten für die Erstellung eines Blocks steigt das Coin Age rapide an und die Anzahl der erstellten Blöcke steigt stark, was wiederum das Coin Age erhöht. Dieser Effekt ist natürlich verstärkt, je weniger Netzwerkteilnehmer im Netzwerk tätig sind, was den starken Abbau von Proof of Stake bei den späteren Experimenten teilweise erklärt. Generell scheint die Anzahl erstellter Blöcke bei allen Verfahren in den späteren Experimenten leicht zu sinken. Dabei handelt es sich um ein merkwürdiges Phänomen, dass eventuell mit der zunehmenden Auslastung der Rechenkapazitäten zu tun hat. Wie bereits kurz beschrieben könnte hier ein weiteres Experiment auf einem tatsächlichen Sensornetzwerk helfen diese Hypothese zu bestätigen, oder andere Gründe für das Problem offenlegen.
Die anderen drei Consensus Verfahren verhalten sich mehr oder weniger wie erwartet. Proof of Work und Distributed Proof of Work verhalten sich ziemlich ähnlich, während Practical Proof of Kernel Work etwas weniger Blöcke erstellt, was sich leicht durch die komplexeren Zugangsbeschränkungen erklären lässt. Ebenfalls gut ersichtlich wird die Verteilung des Minings auf das ganze Netzwerk. Jeder einzelne Netzwerkteilnehmer erstellt relativ wenige Blöcke, während die Blockchain im Vergleich relativ lang ist. Dies deutet auf einen hohen Grad an Sicherheit und hohe Fairness beim Mining hin. Es sollte für einen Angreifer also entsprechend schwer sein große Mengen der Blockchain selbst zu erstellen und damit den Datenstand zu manipulieren.
Entsprechend zum Vergleich sind in den folgenden Abbildungen 5.10 bis 5.13 noch einmal die besagten Zeiten zwischen den erfolgreichen Miningvorgängen beziehungsweise den Blöcken der Blockchain graphisch dargestellt. Aus den Werten lässt sich leicht erkennen, dass die Zeiten zwischen den einzelnen Miningvorgängen eines Netzwerkteilnehmers sehr hoch werden können. Es kann also keinesfalls auf den Broadcast von Transaktionen im Netzwerk verzichtet werden, da diese Daten ansonsten sehr lange nicht Teil der Blockchain werden und den anderen Netzwerkteilnehmern somit nicht zur Verfügung stehen, um sie in ihre Analyse mit einzubeziehen. Durch die großen Unterschiede zwischen der Verfahren können die Mining Times der einzelnen Netzwerkteilnehmer stark variieren. Der regelmäßige Abgleich der Blockchain hält die Zeit zwischen den Blöcken jedoch auf einem ähnlichen Niveau. Generell scheint die Zeit zwischen den Blöcken der Blockchain zumindest bei wenigen Netzwerkteilnehmern aber in dem Bereich zu liegen, der innerhalb der Implementierung festgelegt wurde, also um die 40 Sekunden. Etwaige Schwankungen in den Werten, lassen sich durch den hohen Zufallsfaktor des Minings erklären. Hier könnte man durch einen Versuch über einen größeren Zeitraum hinweg Abhilfe schaffen.

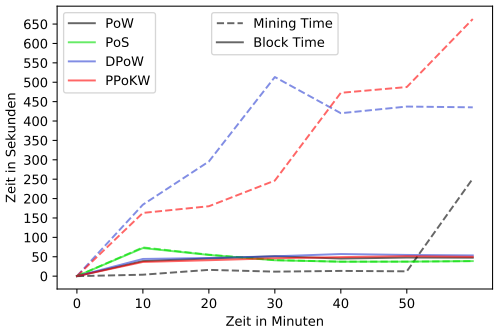

Abbildung 5.10: Durchschnittliche Mining Time/ Block Time bei 5 Netzwerkteilnehmern über 1 Stunde

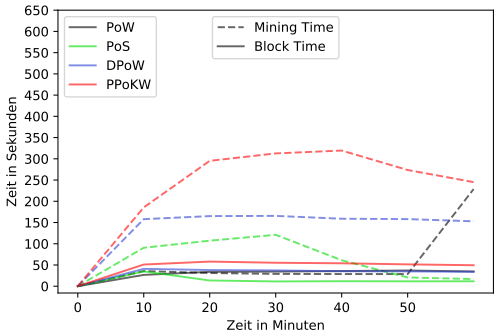

Abbildung 5.11: Durchschnittliche Mining Time/ Block Time bei 10 Netzwerkteilnehmern über 1 Stunde

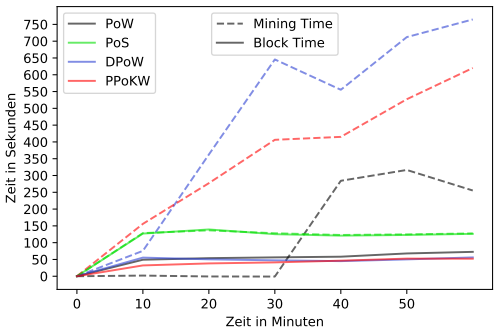

Abbildung 5.12: Durchschnittliche Mining Time/ Block Time bei 20 Netzwerkteilnehmern über 1 Stunde

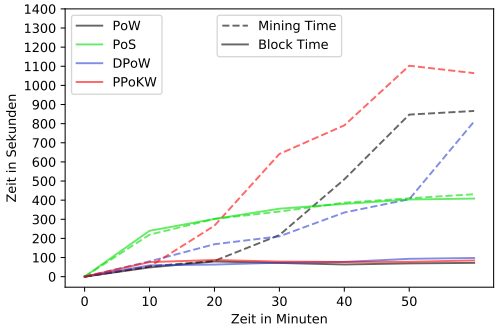

Abbildung 5.13: Durchschnittliche Mining Time/ Block Time bei 40 Netzwerkteilnehmern über 1 Stunde

Auch wenn der Fokus in diesem Teil der Arbeit nicht auf dem Anwendungsfall liegen soll, sollen hier einige Zahlen zum Geometric Monitoring genannt werden. Die Tabelle stellt die durchschnittliche Entdeckungszeit von Grenzwertüberschreitungen in Sekunden, ab dem Eingang der Daten, für die 16 Experimente dar.

Netzwerkteilnehmer	PoW	PoS	DPoW	PPoKW
5	37.39	43.19	32.68	31.55
10	63.13	67.96	43.04	39.22
20	176.82	174.75	98.69	80.76
40	696.01	657.51	368.62	342.19

Diese Tabelle soll nur als zusätzliche Ergänzung zu den in den Abbildungen dargestellten Informationen fungieren. Um Schlüsse über die Effizienz von Geometric Monitoring auf Blockchains zu ziehen, wäre es nötig gewesen ausführlichere Informationen zur Datenanalyse zu sammeln. Dies war aber wie bereits beschrieben nicht Ziel der durchgeführten Experimente. Generell lässt sich sagen, dass das Geometric Monitoring auf der Blockchain funktioniert hat und Grenzwertüberschreitungen entsprechend erkannt wurden.

5.3.3 Speicher

In diesem Abschnitt geht es, um den verwendeten Arbeitsspeicher der verschiedenen Verfahren. Dazu wurden in Python jeweils für jeden Netzwerkteilnehmer der Speicherverbrauch seines jeweiligen Prozesses gemessen. Die Ergebnisse finden sich in den Abbildungen 5.14 bis 5.17.

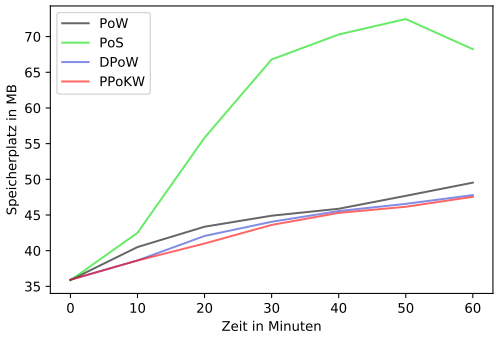

Abbildung 5.14: Durchschnittlicher Speicherverbrauch bei 5 Netzwerkteilnehmern über 1 Stunde

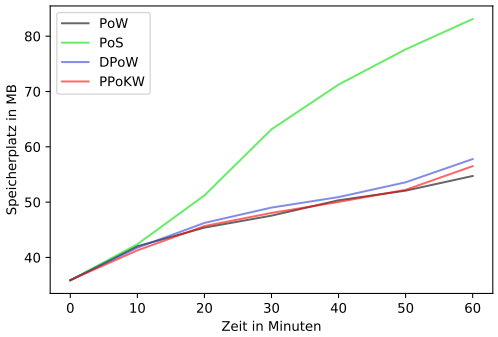

Abbildung 5.15: Durchschnittlicher Speicherverbrauch bei 10 Netzwerkteilnehmern über 1 Stunde

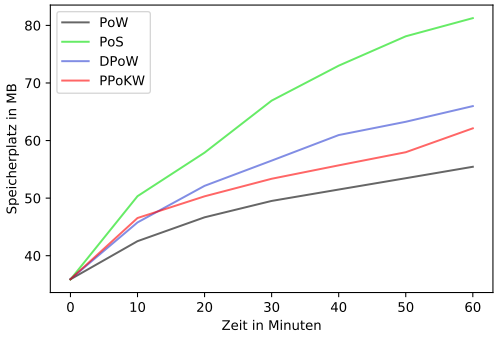

Abbildung 5.16: Durchschnittlicher Speicherverbrauch bei 20 Netzwerkteilnehmern über 1 Stunde

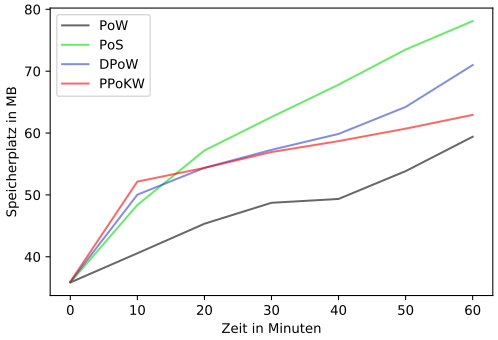

Abbildung 5.17: Durchschnittlicher Speicherverbrauch bei 40 Netzwerkteilnehmern über 1 Stunde

Der verwendete Arbeitsspeicher scheint bei allen vier Verfahren in einem sehr ähnlichen Bereich zu liegen. Wenig überraschen benötigt Proof of Work am wenigsten Arbeitsspeicher, während Pracitcal Proof of Kernel Work und Distributed Proof of Work aufgrund ihrer zusätzlichen Zugangsbeschränkungen etwas mehr Speicher benötigen. Eine sehr große Rolle spielt hier wohl auch die Verifiable Random Function, die in beiden Verfahren implementiert ist. Proof of Stakes erhöhter Arbeitsspeicherverbrauch resultiert vermutlich aus der Berechnung des Coin Ages, die die Komplexität des Verfahrens enorm erhöht. Dieser Effekt wird verstärkt, um so mehr Transaktionen in der Blockchain enthalten sind, wodurch sich die Entwicklung in den späteren Experimenten erklären lässt.

5.3.4 Rechenleistung

Im letzten Abschnitt zur Analyse der Ergebnisse soll es um die Rechenleistung gehen. Zu diesem Zweck wurden erneut in Python Systeminformationen ausgelesen. Die in den Abbildungen 5.18 bis 5.21 dargestellte Prozentzahl beschreibt dabei die Auslastung des aktuellen CPU-Kerns, auf dem der Prozess läuft.

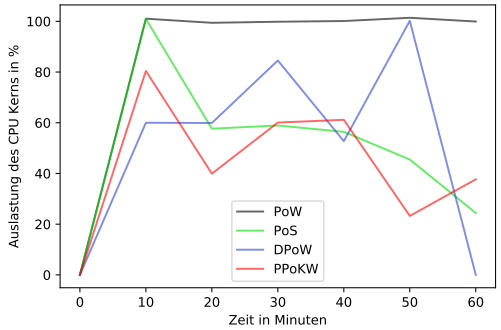

Abbildung 5.18: Durchschnittliche CPU-Auslastung bei 5 Netzwerkteilnehmern über 1 Stunde

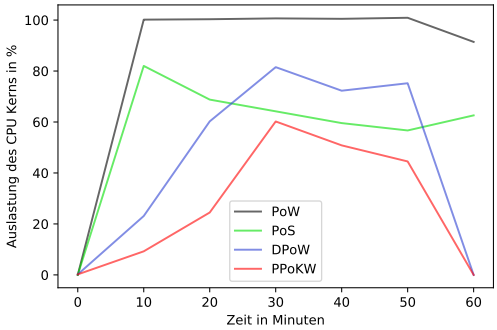

Abbildung 5.19: Durchschnittliche CPU-Auslastung bei 10 Netzwerkteilnehmern über 1 Stunde

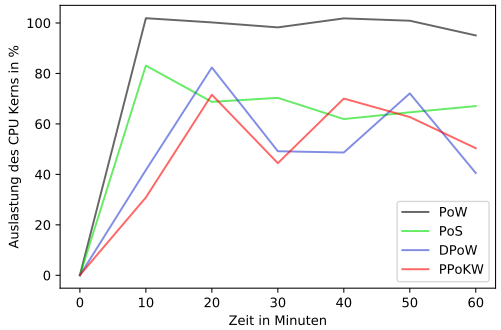

Abbildung 5.20: Durchschnittliche CPU-Auslastung bei 20 Netzwerkteilnehmern über 1 Stunde

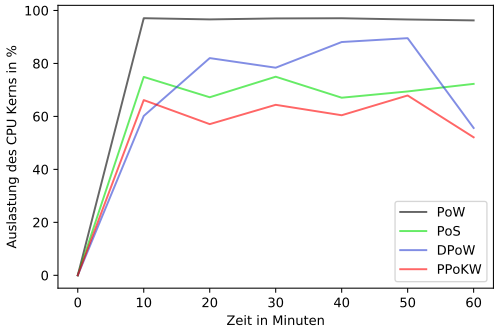

Abbildung 5.21: Durchschnittliche CPU-Auslastung bei 40 Netzwerkteilnehmern über 1 Stunde

Bei diesem Ergebnis wird der größte Nachteil von Proof of Work offensichtlich. Es lastet zu nahezu jeder Zeit den jeweiligen CPU-Kern vollständig aus. Alle anderen Verfahren, deren erklärtes Ziel es war den Ressourcenverbrauch von Proof of Work zu reduzieren, gelingt es die CPU weniger auszulasten. Besonders Pracitcal Proof of Kernel Work und Distributed Proof of Work stechen hier hervor. Durch die Mining Races gibt es zu jeder Zeit CPUs, die nicht vollständig ausgelastet werden, sodass im Schnitt deutlich weniger CPU Leistung verbraucht wird. Proof of Stakes Erfolg ist einfach durch die geringere Mining Difficulty zu erklären. Generell scheint die Auslastung der CPU auch weitgehend unabhängig von der Größe des Netzwerks zu sein und ist damit sehr skalierbar.

5.4 Zusammenfassung und Fazit

In dieser Arbeit wurde die Technologie Blockchain in verschiedenen Varianten analysiert, um die Frage beantworten zu können, ob Blockchains ein Möglichkeit zur dezentralen Datenanalyse auf Netzwerken heterogener Geräte darstellen. Dazu wurde zunächst einmal ein Überblick über die Blockchain und die aktuelle Forschungsarbeit in diesem Bereich gegeben. Es wurden die vier zentralen Varianten der Blockchain herausgearbeitet und ihre jeweiligen Besonderheiten herausgestellt.
Proof of Work, das Verfahren mit dem die Blockchain ursprünglich entwickelt wurde, ist sicherlich das verbreitetste Consensus Verfahren und bekannt für seine Einfachheit aber auch seinen enormen Ressourcenverbrauch. Das nächste Verfahren war Proof of Stake, welches aus Proof of Work entwickelt wurde, um eben diesen Ressourcenverbrauch zu reduzieren. Hier lag der Fokus auf der Reduzierung der Mining Difficulty. Die letzten beiden Varianten Distributed Proof of Work und Practical Proof of Kernel Work sind sich sehr ähnlich und setzten bei einer Zugangsbeschränkung zum eigentlichen Proof of Work an. Dabei verwendet Distributed Proof of Work lediglich eine Zufallsauswahl der Miner, während Pracitcal Proof of Kernel Work zusätzlich noch eine Whitelist und eine Menge aus dynamischen Regeln verwendet.
Nachdem die möglichen Chancen und Probleme der Technologie analysiert wurden, war der nächste Schritt eine Implementierung. Der Fokus lag hier auf Einfachheit und einer Reduzierung auf die Besonderheit der einzelnen Verfahren. Als Programmiersprache wurde Python gewählt, um sich die Möglichkeit auf eine Umstellung auf MicroPython offen zu halten.
Als nächstes wurden kurz ein Anwendungsfall sowie das Datenanalyseverfahren Geometric Monitorig erläutert, um dann eine Reihe von Experimenten durchführen zu können, welche mehr Aufschluss über die Eignung von Blockchains in Datenanalyse Szenarien geben sollten. Für die Experimente wurden Daten von einem frei verfügbaren Sensornetzwerk namens openSenseMap verwendet. Die Ergebnisse wurden unter den vier verschiedenen Gesichtspunkten Kommunikation, Mining, Speicher und Rechenleistung betrachtet und entsprachen größtenteils den Erwartungen. Kommunikationskosten scheinen ein großes Problem zu sein und auch bei der Skalierbarkeit kann es in vielen Bereichen zu Problemen kommen. Generell ist eine Analyse von Daten auf einer Blockchain aber durchaus möglich.

Es ist schwer zum jetzigen Zeitpunkt ein abschließendes Urteil über Blockchains zu fällen. Eines der zentralen Themen scheint der Datendurchsatz zu sein. Für Anwendungsfälle, die in einem geringen Turnus neue Daten liefern sollten durch eventuell länger andauernde Mining Prozesse keine Probleme entstehen. Anwendungsfälle mit einer hohen Frequenz an Daten laufen jedoch Gefahr, dass die Analyse einzelner Datenpunkte sehr lange dauern kann. Generell scheint eine gute Abstimmung der durchschnittlichen Zeit zwischen Blöcken

auf den Erhalt neuer Werte eine zentrale Rolle zu spielen.
Ein ganz anderes und zu diesem Zeitpunkt ungelöstes Problem sind die enormen Kommunikationskosten, die sich negativ auf die Skalierbarkeit auswirken. Es ist dringend notwendig Wege zu finden, um die Kommunikation im Netzwerk zu optimieren, wenn die Blockchain langfristig Erfolg auf heterogenen Geräten haben soll.
Bei der Rechenleistung hingegen wurden große Fortschritte gemacht. Distrbuted Proof of Work und Practical Proof of Kernel Work ist es gelungen das zentrale Problem von Proof of Work deutlich zu reduzieren, ohne dabei die Sicherheit und die Performanz der Blockchain deutlich zu verschlechtern. Beide Verfahren sind vielversprechende Kandidaten, die mit etwas Optimierung vermutlich zur Datenanalyse auf Blockchains eingesetzt werden können. Mit dem aktuellen Stand an Informationen ist von dem Einsatz von Proof of Work und Proof of Stake zur Datenanalyse im Speziellen auf heterogenen Geräten abzuraten. Auch wenn Proof of Stake durchaus Potenzial für Optimierungen besitzt, welche es sogar geeigneter als die anderen beiden Consensus Verfahren machen könnte.
Ein großer Pluspunkt der Blockchain bleiben natürlich die Sicherheit und die Ausfallsicherheit, was gerade in Anwendungsfällen, die anfällig für Manipulationen und Ausfälle sind, das ausschlaggebende Argument sein kann. Hier liegt vermutlich das größte Potenzial der Blockchain, sodass sie in diesem Bereich am ehesten mit alternativen Technologien konkurrieren kann. Es ist definitiv notwendig weitere Forschungsarbeit in diesem Themengebiet durchzuführen, um zu einem abschließenden Urteil zu kommen. Einige Ansätze dazu finden sich auch im folgenden Ausblick.

Kapitel 6

Ausblick

Im Folgenden geht es darum, auf einige offen gebliebene Fragen einzugehen und Anregungen für weiterführende Arbeiten zu diesem Thema zu schaffen.
Die im Rahmen dieser Arbeit durchgeführten Experimente und ermittelten Ergebnisse sind keinesfalls erschöpfend. Wie auch in den Beschreibungen der Consensus Verfahren und besonders in der Analyse dieser schon herausgestellt gibt es etliche Stellschrauben innerhalb der Datenstruktur Blockchain, die feinjustiert werden können, um bestimmte Ziele zu erreichen. Diese hängen vor allem von dem letztendlichen Anwendungsfall ab, für welchen die Blockchain eingesetzt werden soll. Um nur einige wenige Beispiele zu nennen, kann zum Beispiel durch eine Veränderung der Komplexität des im Proof of Work zu lösenden Problems die Anzahl erstellter Blöcke und damit der Datendurchsatz des Systems enorm erhöht werden. Ein weiterer Ansatz ist die Feinjustierung der Verifiable Random Functions, um ein dem Anwendungsfall angemessenes Verhältnis von Sicherheit und Komplexität zu finden. Es wäre dementsprechend wünschenswert zusätzliche Reihen von Experimenten durchzuführen, um weitere Statistiken über derartige Konfigurationen zu erhalten.
Ein weiteres zentrales Thema ist die Netzwerkstruktur. Die aktuelle Struktur ist zwar sehr übersichtlich und zuverlässig, setzt aber die Speicherung der Adressen aller Netzwerkteilnehmer voraus und stellt damit eine große Herausforderung für die Skalierbarkeit des Netzwerks dar. Langfristig gilt es also die Implementierungen auf P2P-Netzwerke wie zum Beispiel Kademlia [13] umzustellen. Diese verfügen über deutlich reduzierte Routingtabellen und können damit auch bei sehr großen Mengen an Netzwerkteilnehmern effizient kommunizieren. Dafür bringen sie wiederum ganz neue Probleme mit sich. So sind zum Beispiel Broadcasts von Nachrichten in P2P-Netzwerken keinesfalls trivial und mit entsprechenden Tücken verbunden. Gerade in Zusammenhang mit dieser Thematik könnten auch die Auswirkungen untersucht werden, die entstehen sollten Netzwerkteilnehmer Transaktionen nur an Teile des Blockchain Netzwerks senden. Wie hoch ist die Wahrscheinlichkeit, dass in einem solchen Fall Transaktionen verloren gehen? Macht es Sinn die Transaktionen über mehrere Blockzyklen hinweg zu senden? Wie schlimm ist der Verlust einzelner

Transaktionen für den Datenverarbeitungsprozess? Auch diese Fragen sind natürlich wieder stark mit dem jeweiligen Anwendungsfall verknüpft.
Ein weiteres sehr interessantes Thema, auf das zuvor bereits mehrfach verwiesen wurde, sind die Investmentstrategien innerhalb von Proof of Stake. So stellt sich zum Beispiel zunächst einmal die Frage, ob es je nach Anwendungsfall Sinn ergibt die Netzwerkteilnehmer mit gewissen Mengen an Kapital zu initialisieren. Diese Initialisierung könnte im Fall von heterogenen Geräten auch uneinheitlich sein und direkt von den Parametern des jeweiligen Geräts abhängen, um so eventuelle Ungleichheiten beim Mining zu minimieren.
Ebenfalls bereits angeschnitten wurde die mögliche Umstellung der Implementierung auf Micro Python. Dies könnte erheblich dabei helfen die begrenzten Ressourcen der heterogenen Geräte noch besser zu nutzen. Dieses Vorhaben hängt natürlich stark von der Entwicklung der Codebasis von Micro Python ab. Generell gilt, dass die in der Arbeit beschriebenen Verfahren keinesfalls an Python oder Python ähnliche Programmiersprachen gebunden sind. Auch Implementierungen in anderen Sprachen könnten neue Einblicke in das Thema bringen. Besonders interessant wären hier natürlich auch Sprachen, die nah an der Hardware arbeiten.
Weiterhin bietet es sich an verschiedene Datenanalyse Verfahren auf der Blockchain zu testen. Im Rahmen der Recherche für diese Arbeit hat sich das in [7] beschriebene Verfahren zur Parallelisierung von Machine Learning mit Hilfe von Radon Points als vielversprechend dargestellt, wurde dann allerdings zu Gunsten von Geometric Monitoring zunächst einmal zurückgestellt. Es wäre also durchaus interessant dieses Verfahren einmal innerhalb der verschiedenen Consensus Verfahren zu implementieren. Dies gilt natürlich entsprechend für alle weiteren Verfahren zur Analyse von Daten, für die man sich einen Mehrwert aus der Technologie Blockchain verspricht.

Anhang A

Weitere Informationen

A.1 Verwendete Python Pakete

Es folgt eine kurze Auflistung der für die Implementierung verwendeten Python Pakete inklusive einer kurzen Beschreibung. Dabei werden nur Pakete genant, die nicht Teil der Python Standard Pakete sind:

Paketname	Beschreibung
Requests	Versenden von http-Requests
Pandas	Datenverwaltung, Import und Export von csv-Dateien
Jsonpickle	JSON De- und Encoding
Psutil	Zugriff auf Systemressourcen
Flask	Aufsetzten von Webservern

A.2 Durchführung eines Experiments

A.2.1 Benötigte Dateien

Es werden folgende Dateien für die Durchführung eines Experiments benötigt:

Dateiname	Beschreibung
data.csv	Enthält die einzulesenden Daten
blockchain_pow.py	Implementierung einer Blockchain mit PoW
blockchain_pos.py	Implementierung einer Blockchain mit PoS
blockchain_dpow.py	Implementierung einer Blockchain mit DPoW
blockchain_pokw.py	Implementierung einer Blockchain mit PPoKW
run_test.py	Skript mit http-Requests zur Inbetriebnahme der Blockchain
run_test2.py	Skript zur Initialisierung des Netzwerks

A.2.2 Arbeitsumgebung einrichten

Als nächstes muss die Python Arbeitsumgebung eingerichtet werden. Im Rahmen dieser Arbeit wurde die Python Version 3.6.6 verwendet. Die benötigten Pakete können aus der obigen Tabelle entnommen werden. Es empfiehlt sich die Verwendung eine Paketverwaltungssoftware wie zum Beispiel Anaconda [1].

A.2.3 Experiment starten

Mit den folgenden Schritten lässt sich das Experiment starten:

1. Sicherstellen, dass alle benötigten Dateien im richtigen Verzeichnis sind
2. Python Terminal starten und folgenden Befehl eingeben
3. `python run_test2.py -n` [`Anzahl an Netzwerkteilnehmern`]
   `-f` [`Datei mit dem gewünschten Verfahren`] `-t` [`Zeit in 10 Minuten Schritten`]
4. Nach der angegebenen Zeit findet sich das Ergebnis in der Datei Logfile.txt

Abbildungsverzeichnis

Algorithmenverzeichnis

Literaturverzeichnis

[1] Anaconda: *Anaconda: Python Paketverwaltung.* Web: `https://www.anaconda.com/`. zuletzt abgerufen am 31.03.2019.

[2] Buterin, Vitalik et al.: *A next-generation smart contract and decentralized application platform.* white paper, 2014.

[3] CICADA: *Cicada: A Distributed Direct Democracy and Decentralized Appilcation Platform.* Web: `iamcicada.com/whitepaper/`. zuletzt abgerufen am 30.03.2019.

[4] Flymen, Daniel van: *HackerNoon: Learn Blockchains by Building One.* Web: `https://hackernoon.com/learn-blockchains-by-building-one-117428612f46`. zuletzt abgerufen am 02.04.2019.

[5] Goldberg, Sharon, Moni Naor, Dimitrios Papadopoulos und Leonid Reyzin: *NSEC5 from Elliptic Curves: Provably Preventing DNSSEC Zone Enumeration with Shorter Responses.* IACR Cryptology ePrint Archive, 2016:83, 2016.

[6] Institut für Geoinformatik Münster: *openSenseMap.* Web: `https://opensensemap.org`. zuletzt abgerufen am 30.03.2019.

[7] Kamp, Michael, Mario Boley, Olana Missura und Thomas Gärtner: *Effective Parallelisation for Machine Learning.* In: *Advances in Neural Information Processing Systems*, Seiten 6477–6488, 2017.

[8] Keren, Daniel, Guy Sagy, Amir Abboud, David Ben-David, Assaf Schuster, Izchak Sharfman und Antonios Deligiannakis: *Geometric monitoring of heterogeneous streams.* IEEE Transactions on Knowledge and Data Engineering, 26(8):1890–1903, 2014.

[9] King, Sunny und Scott Nadal: *Ppcoin: Peer-to-peer crypto-currency with proof-of-stake.* self-published paper, August, 19, 2012.

[10] Lazerson, Arnon, Daniel Keren und Assaf Schuster: *Lightweight monitoring of distributed streams.* ACM Transactions on Database Systems (TODS), 43(2):9, 2018.

[11] Lundbæk, Leif-Nissen, Daniel Janes Beutel, Michael Huth und Laurence Kirk: *Practical Proof of Kernel Work & Distributed Adaptiveness*. Manuscript Version 1.2, 2018.

[12] May, Michael, Bettina Berendt, Antoine Cornue et al.: *Research challenges in ubiquitous knowledge discovery*. In: *Next Generation of Data Mining*, Seiten 154–173. Chapman and Hall/CRC, 2008.

[13] Maymounkov, Petar und David Mazieres: *Kademlia: A peer-to-peer information system based on the xor metric*. In: *International Workshop on Peer-to-Peer Systems*, Seiten 53–65. Springer, 2002.

[14] Micali, Silvio, Michael Rabin und Salil Vadhan: *Verifiable random functions*. In: *Foundations of Computer Science, 1999. 40th Annual Symposium on*, Seiten 120–130. IEEE, 1999.

[15] Milutinovic, Mitar, Warren He, Howard Wu und Maxinder Kanwal: *Proof of luck: An efficient blockchain consensus protocol*. In: *Proceedings of the 1st Workshop on System Software for Trusted Execution*, Seite 2. ACM, 2016.

[16] Nakamoto, Satoshi: *Bitcoin: A peer-to-peer electronic cash system*. 2008.

[17] P4Titan: *Slimcoin: A peer-to-peer crypto-currency with proof-of-burn*. 2014.

[18] Sharfman, Izchak, Assaf Schuster und Daniel Keren: *A geometric approach to monitoring threshold functions over distributed data streams*. ACM Transactions on Database Systems (TODS), 32(4):23, 2007.

[19] Wood, Gavin: *Ethereum: A secure decentralised generalised transaction ledger*. Ethereum project yellow paper, 151:1–32, 2014.

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet sowie Zitate kenntlich gemacht habe.

Dortmund, den 3. April 2019

Cedric Sanders